SCORING THE WRONG QUESTION: READOUT FAILURES IN CONSTRAINED-OPTION EVALUATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Constrained-option scoring reads a model's probabilities for a fixed set of permitted answers, so it returns a score even when the model is about to write something else. We study a prompt that quotes a multiple-choice item, ending in the item's own answer instruction, and asks for a forecast of whether a reader model given only a short note will answer it correctly. The model can begin answering the quoted item: on three Qwen3 checkpoints an answer letter is the most probable token on nearly every passage, and the forecast ranks the reader's correctness indistinguishably from chance. Beyond prior work's argmax check and prefill repair, we contribute a label-free diagnostic of the declared options' support, their first-token mass in context; a deletion test tracing the Qwen3 answer-letter start to the quoted instruction; and rendering and tokenisation checks for dropped separators and coinciding first tokens. Prefilling an answer stem lifts the median option mass from below to above one half on all nine checkpoints scored with this prompt and, with options and renormalisation unchanged, raises the Qwen3-averaged AUC from 0.4929 to 0.6152. lm-polygraph's default P(True) estimator shows a related failure: it reads True where Qwen3 without thinking favours an answer letter (MMLU) or the prompt's (A)/(B) label (TriviaQA), and its Qwen3-averaged AUROC is indistinguishable from chance on MMLU and below chance on TriviaQA. Scoring and renormalising the (A)/(B) labels after an answer stem raises it to 0.632 and 0.868, renormalising True against False in place reaches 0.660 and 0.860, and on TriviaQA both exceed all fourteen default single-answer library estimators. The original forecast is renormalised too, yet stays indistinguishable from chance: option mass flags low support in both settings without labels, and only checking the intended target shows which score still ranks.

1 INTRODUCTION

A model can answer a quoted question while an evaluation reports its forecast of someone else's answer. In Figure 1, the prompt quotes a reading-comprehension item with a letter-answer instruction, then asks how many of a hundred readers would answer correctly. The model answers the item first; the scorer reads that position and returns a distribution over its permitted numbers.

At the original position, the first token of the reply, the forecast ranks reader correctness indistinguishably from chance: its AUC averaged over three Qwen3 checkpoints is 0.4929 $[0.4390, 0.5458]$. On 240 shared passages, an answer stem, text prefilled at the start of the reply, raises the Qwen3-averaged AUC to 0.6152 and Meta-Llama-3-8B-Instruct's from 0.5231 to 0.6044. Table 4 gives paired intervals and Brier scores; Section 6 compares the stemmed Brier scores with a constant 0.5 forecast.

How the scoring hides the failure. Constrained-option scoring compares a fixed set of legal continuations, the *declared set*, at one next-token position, the *scored position*. Renormalising over that set discards all other probability mass. Our declared set is the forecast grid $\{10, 30, 50, 70, 90\}$, whereas the quoted item requests A, B, C or D. Section 4 measures the answer-letter argmax and forecast mass. The unrestricted continuation, the model's greedy output there, reveals which instruction its first token follows.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Rendered prompt ending

```
After the woman visited the United States
↪  she might feel that   _  .
A: her English was poor
B: it's easy to master English
C: it's difficult to make money
D: people there weren't very friendly
Answer with exactly one of: A, B, C, D.

Out of 100 such readers, how many answer
↪  correctly? Reply with exactly one of:
↪  10, 30, 50, 70, 90.<|im_end|>
<|im_start|>assistant
```

Scored here: the next token after the final newline.

Next-token distribution

Token	Probability
A	**1.0**
Answer	3.36×10^{-10}
D	3.28×10^{-11}
After	8.29×10^{-12}
5	5.27×10^{-12}
The	4.73×10^{-12}
Five-option mass	5.36×10^{-12}

Unrestricted continuation

A	answers the item
50	gives the forecast

The exact output is `A  \n50`.

Figure 1: The model answers the quoted item before forecasting. On this real Qwen3-4B item, the note states the answer, A. The later forecast instruction asks for one of five numbers, but the scored position assigns its highest probability to A; with notes about an unrelated passage, which state no answer, 58 of 60 rows still put an answer letter first (Section 4). The table shows the six most probable tokens and the mass of the five options' distinct first tokens. The unrestricted continuation reaches the forecast only after answering the item. Appendix B gives the complete prompt and both scored positions; line wrapping here is for display.

What the experiments establish. Beyond the argmax-membership test and prefill repair of Cappelletti et al. (2025), we add the option-mass diagnostic, localisation to the quoted instruction, tokenisation checks and consequences in maintained evaluation software. Deleting the quoted item's closing instruction removes the answer-letter start on the three Qwen3 checkpoints and, at the original ending, lowers its rate to 0.016 on SmolLM2 and Mistral; Phi-3 needs the choice list removed too. Deleting only the choice list leaves an answer-letter argmax on 67 to 89% of items on the three Qwen3 checkpoints and SmolLM2 (Appendix D). Substituted instructions also change the generated response, but the tested token-level rules do not explain how. A judge and human labels assess instruction execution from the continuation (Section 4). Section 5 separates answer-letter starts from missing prompt separators across seven checkpoints in five architecture families. lm-polygraph's P(True) confidence ranks correctness indistinguishably from chance on MMLU (Hendrycks et al., 2021) and below chance on TriviaQA (Joshi et al., 2017), averaged over three Qwen3 checkpoints without thinking (Section 4). The same design appears in maintained scorers: lm-polygraph enables P(True) in 23 of its 30 configurations, AlpacaEval reads log-probabilities over its declared set in 12 of its 44 annotator configurations, and 27 of 131 OpenCompass LLM-judge grader configurations quote a lettered item and read the judge's first A or B (Appendix E).

First-token readouts also require distinct option tokens after the actual prompt; isolated encoding can score a shared marker. The experiment map, Table 1, gives the samples and readouts; Section 6 gives the practical checks.

2 Related work

An option's probability and the answer it represents. A string can represent an answer without its probability measuring the model's preference for that answer. Holtzman et al. (2021) describe how generic strings absorb probability and leave intended answers in a poorly calibrated tail. Their COPA-Flipped control (Holtzman et al., 2021, Section 5, Table 5) removes the hypothesised cause and the scoring-rule gap disappears; our instruction-deletion control in Section 4 shares that design. Robinson et al. (2023) likewise distinguish sentence probability from answer correctness. Their multiple-choice alternative requires a model to bind a symbol to an option, and they show GPT-3 assigning its highest probability to `A` under both orderings of one list. Our quoted item uses that ability: it has its own labels and instruction, which can take over a position scored only for forecasts. Their repair uses arbitrary symbols; our options are an ordered numeric grid whose ordinal structure is the measurement. Sun et al. (2026, Section 6) show models writing instructions for agents in the wrong roles; here a model reads and executes an instruction addressed to another reader.

Instructions inside quoted material. Greshake et al. (2023, Sections 1 and 3) define indirect prompt injection with an adversary placing instructions in retrieved data. The data–instruction confusion it exploits also occurs without an attacker: Zverev et al. (2025, Sections 1 and 3, Definitions 3–4; Section 4) place a probe in the data and count its one-word answer anywhere in the output; Hwang et al. (2025, Sections 1 and 3.1; Table 3) study rewriting, proofreading, translation and style transfer when the input itself resembles an instruction, such as a question or a coding request. Wallace et al. (2024, Section 1, Figure 1; Section 3.2) assign privilege by message type. In our scoring prompt, the quoted item's instruction and the scorer's question share one user turn, so we infer that this hierarchy assigns them the same privilege. These studies read generated text; we read the scorer's side, where answering the quoted item first is enough to put its answer letter at the scored position.

Prompt format and the position being scored. Sclar et al. (2024) require instructions and options to match: changing option characters also changes every reference to them in the instruction. Our prompt contains two different requested option sets. Their option-ranking metric omits probability outside the options. Both their study and Robinson et al. (2023) measure whitespace effects. Sanz-Guerrero et al. (2025) report whitespace-related accuracy changes up to 11% and changes in model rankings. They recommend tokenising the space with the answer letter; for numeric labels their two-element split uses the final element, where digits remain distinct, rather than the shared first element.

Wang et al. (2024b) directly compare first-token scores with generated text. Even their instruction to start with a single letter leaves misalignment. Their subject is agreement between readouts; ours is a declared set that excludes what sits at the scored position. Wang et al. (2024a) compare which readout survives perturbation. An answer stem instead supplies text in the assistant response, the distinction studied by Cappelletti et al. (2025). Karim et al. (2026) distinguish failure to enter answer mode from failure to select the right candidate once in that mode, using a log-probability margin over candidates. Our case is their gating failure: the model's next token is not among the scored candidates. We use the existing prefill mechanism and identify two misleading measurements from encoding options in isolation (Section 6; Appendix C.5).

Calibration and diagnostic checks. Calibration and selection-bias methods operate within the scored option set and assume that its position is correct (Zhao et al., 2021; Zheng et al., 2023); theirs is a skew within that set, ours mass outside it. Numeric labels themselves work as a readout: Zheng et al. (2023) report MMLU accuracy of 65.8 with numeric labels and 67.2 with letters on `gpt-3.5-turbo`. For checking metrics, Shankar et al. (2024) combine coverage and false failure rate for boolean assertions. Sai et al. (2021) pair changes a continuous metric must detect with invariant cases whose score must not drop. Cappelletti et al. (2025) measure the unrestricted highest-probability token's membership in the option set across examples. Restricting the argmax to the set would remove membership as a diagnostic while leaving option mass measurable.

3 TASK, MEASUREMENTS AND EXPERIMENTS

A writer model compresses a passage into a short note under a word budget. A reader model sees only the note and answers the passage's multiple-choice question. The forecasting model sees the note and the complete item and selects from $\mathcal{O} = \{10, 30, 50, 70, 90\}$: how many of a hundred such readers would answer correctly? The quoted item contains its question, lettered choices and closing instruction. We call this unchanged forecast template, rendered with the checkpoint's chat template and without an answer stem, the *production prompt*; its *original position* is the next token after the assistant turn's opening. The *answer stem* `"Out of 100 such readers, about "`, ending in a space, is appended after that opening; the *stemmed position* is the next token after it. `lm-evaluation-harness` (Gao et al., 2026) provides such a response prefix (Appendix F). In the downstream task, one checkpoint writes the note, reads it as the note-only reader and forecasts; the target is that one reader's correctness, and "Out of 100 such readers" is the question's wording.

Items come from RACE-H (Lai et al., 2017) passages of at least 200 words, with one question per passage. Every RACE-H sample is a prefix of the same shuffled list. Appendices A.1 and H specify sampling and note budgets. The main text uses three RACE-H samples: 64 items with one fixed note (Table 3, Sections 4 and 5); 120 rows on Qwen3-4B with constructed notes, half stating the answer and half about an unrelated passage (Section 4); and 240 passages with the notes each checkpoint

writes itself (Table 4). Appendix I also tests notes written for individual items. The text shortens checkpoint names, for example to Phi-3 for `Phi-3-mini-4k-instruct`; Table 3 gives the full names. The judge reads the same first 32 items across all 49 checkpoint–instruction combinations; the experiment map lists the other samples and readouts. Intervals are 95% percentile bootstrap intervals, and a Qwen3-averaged value is the mean of the three Qwen3 checkpoints' values.

Read the whole next-token distribution. Let c be the prompt after chat-template rendering, through the start of the assistant turn. At the scored position, $p(t \mid c)$ is the probability of the next token t. We record its unrestricted argmax as well as the *in-context first-token mass*, or *option mass*, $\mu(c)$: the sum of probabilities of the options' distinct first tokens when each option is encoded as a continuation of c. Each token is counted once, even if several options share it. This mass upper-bounds the total probability of the full option sequences; Appendix A gives the formula and its assumptions. The *answer-letter rate* is the fraction of items whose argmax is an A–D letter.

We call a median option mass below the diagnostic floor of 0.5 *collapse*. On an item at or above the floor, at least half the next-token probability falls on the options' first tokens; the unrestricted argmax shows whether the most probable next token is one of them. Whether a score over tokens the model is not about to emit ranks a target is a separate, measured question: it does for lm-polygraph's P(True) renormalised over True and False (Section 4); the forecast renormalised at its original position is indistinguishable from chance (Table 4). Shared first tokens cannot distinguish the options, even at high mass. The *unrestricted continuation*, the greedy output there, shows what the model writes.

The forecast that is evaluated downstream. Each checkpoint greedily writes its own note of at most 32 words. As the note-only reader, it picks the option among `" A"`, `" B"`, `" C"` and `" D"` with the highest full-sequence log-probability; matching gold defines one deterministic outcome per passage, and the forecast is evaluated as a probability of that outcome. Appendix H gives the prompts and counts of correct reader answers. On these passages and the three Qwen3 checkpoints, the spaced and unspaced strings select the same option on 96.25% to 98.33% of rows, and the unspaced pick equals the reader's greedy answer on every row where one is parsed; scored against either, the Qwen3-averaged gain of Section 6 is 0.1174 $[0.0382, 0.1969]$ and 0.1163 $[0.0371, 0.1956]$ (Appendix C.7). The paired comparison in Section 6 uses each full option's log-probability ℓ_o to obtain a probability of reader correctness:

$$p_S = \frac{\sum_{o \in \mathcal{O}} (\mathrm{int}(o)/100) \exp(\ell_o)}{\sum_{o \in \mathcal{O}} \exp(\ell_o)}. \tag{1}$$

Here $\mathrm{int}(o)$ is the option's number. Renormalisation gives a weighted mean even when the options' total probability is small. We evaluate ranking with AUC and squared probability error with Brier score; Appendix H specifies encoding and the paired comparison.

4 Which instruction controls the scored position?

The model answers the quoted item first. On Qwen3-4B's 120 constructed-note rows, the unrestricted argmax is an A–D answer letter on 118. None has a forecast option's first token as its argmax. The median option mass is 1.42×10^{-15} (Appendix C.3). Figure 1 shows one of these items: its unrestricted continuation answers the item before forecasting. Greedy continuations of at most 16 new tokens of the production prompt with the fixed note, classed by fixed rules on their opening, answer the item first on 62 or 63 of the 64 items on Qwen3-4B, 8B and 14B and on Phi-3; on those three Qwen3 checkpoints a grid forecast follows within that limit on 61 to 63 of them. Three people's labels agree with the rules on 92 of 100 sampled continuations, 97.9% when reweighted to all 512 on eight checkpoints (Appendix I). This also occurs when the note cannot supply an answer to copy: 58 of the 60 rows with notes about an unrelated passage have answer-letter argmaxes (Appendix B).

Deleting the item's instruction. The quoted item ends with `Answer with exactly one of: A, B, C, D.`; the forecast instruction comes after it. We remove or replace the item's instruction on the same 64 items, holding the note fixed, and also test deletion of its lettered choice list. Deletion is measured at the original ending on all seven checkpoints. Substitution uses that ending on five; on Mistral and falcon, it uses an appended space to supply the separator their templates omit.

Table 1: **Experiment map.** Rows are numbered for reference; n is per checkpoint and condition unless specified otherwise. "Seven" refers to Table 3; "eight" adds Llama-3-8B-Instruct; "nine" also adds Qwen3-32B; "ten" adds Flan-T5-XL, Flan-T5-XXL and Flan-UL2 to the seven; "eleven" also adds GPT-2.

	What varies	Checkpoints	n	What is read	Reported in
1	Note states the answer / unrelated note	Qwen3-4B	120 rows, 60 each	Mass; unrestricted argmax	§4, App. C.3
2	Checkpoint; answer stem	Nine	64	Mass; argmax; below-floor counts	Table 3, §5
3	Delete / substitute instruction; choices; separator	Seven	64	Letter rates and token classes	§4, App. D
4	Judge and human labels	Seven	32 per ending; 100 labelled by hand	Execution labels and agreement	§4, App. D
5	lm-polygraph P(True): library, renormalised, stemmed, deleted, labels in place	Qwen3-4B, 8B, 14B; Llama-3-8B-Instruct; option mass also on Phi-3, Mistral, falcon, SmolLM2, Qwen2.5-72B-Instruct	500 MMLU; 500 TriviaQA	AUROC; prediction–rejection ratio; option mass; argmax class	§4, App. E
6	Leading spaces in reader options	Qwen3-4B, 8B, 14B	240 passages	Choice agreement; mass; accuracy; forecast AUC	§3, App. C.7
7	Quoted item in the BEGIN/END fence; answer stem	Qwen3-4B	64	Option mass and argmax	§6, App. C.4
8	Isolated / in-context encoding	Seven	64	Mass; token distinctness	§6, App. C.1
9	Option strings; scored context	Eleven; tokenisers only	64 contexts	Distinct first tokens	§6, App. C.6
10	Published option sets; scored positions	Ten	64 requested	Mass; option-token argmax	§6, App. F
11	G-Eval score; rating dimension	Seven	100 documents; 1600 rows	Correlation with human ratings	§6, App. G
12	Answer stem; item's instruction deleted; another template; the generated number	Qwen3-4B, 8B, 14B; Llama-3-8B-Instruct	240 passages	Forecast AUC and Brier	§1, §6, App. H
13	Notes; task placement; items; instruction wording	Eight	64	Mass; argmax; unrestricted continuation	App. I

At the original ending, deleting the item's instruction reduces the answer-letter rate to 0.000 on the three Qwen3 checkpoints, to 0.016 on SmolLM2, and from 0.469 to 0.016 on Mistral. On these five checkpoints, the answer-letter argmax depends on that instruction. On Phi-3, the rate falls to 0.500 and reaches 0.000 only when the choice list is removed as well. Its forecast options share a first token on both prompts, so only the letter rate is interpretable. Supplying the missing space with the instruction intact also reduces Mistral's rate to 0.016; falcon's remains 0.000. Appendix D gives the deletion and substitution results, including the mass changes at the original ending, where the separator defect remains on Mistral and falcon.

Interpreting generated continuations. Replacing the item's instruction changes the outputs, but two simple rules fail to explain how: matching the instruction's opening word, and predicting the class of its first response token (Appendix D). We instead ask Grok 4.6 to label whether each continuation executes the replacement instruction on the material shown, answers the item, forecasts, or does none of these. The two instruments read the same cells: the continuation begins with the recorded argmax on 1566 of 1568 rows. Under the translation instruction, the opening-word rule reads 0.0000 on all seven checkpoints, while the judge reads execution on six, at rates from 0.2188 to 0.9688; falcon reads 0.0000. A bare number, a continuation that is only a number, is counted as a forecast. Two

judges from other developers, Claude Opus 5.5 and GPT-6-astra, read translation execution on the same six and agree with Grok's execution labels on 1549 and 1532 of the 1568 continuations. At the original ending, all three judges label 31, 32, 31 and 32 of 32 continuations as item answers on Qwen3-4B, 8B, 14B and Phi-3 with the item's instruction intact, and all 32 on each Qwen3 checkpoint as forecasts with it deleted. Three people labelled a separate sample of 100 continuations without seeing the judge's labels; Grok agrees with their majority on execution on all 99 they label unanimously. Table 8 adds a second labelling and the other agreements (Appendix D).

A maintained implementation, lm-polygraph's default P(True) estimator (Fadeeva et al., 2023), reads `True` where the model writes another token (Appendix E). On 500 MMLU items, an A–D letter is the unrestricted argmax on 387, 496 and 500 items on Qwen3-4B, 8B and 14B without thinking. Of these, 179, 215 and 129 are C or D, which the prompt's `(A) True` and `(B) False` labels do not offer. Once the quoted item's instruction is deleted, a letter is the argmax on 0, 0 and 3 items. On 500 TriviaQA items, the library's own labels `(A` and `(B` hold the position on 463 to 500 items per Qwen3 checkpoint. On the three Qwen3 checkpoints, `True` and `False` hold a median mass of at most 1.2×10^{-10} on the 500 MMLU items and 5.1×10^{-8} on the 500 TriviaQA items.

Table 2: **lm-polygraph P(True) readouts.** Qwen3-averaged AUROC for the model's own answer being correct, 500 items per task; Tables 9 and 10 give each checkpoint for Library, Renormalised, Moved and Deleted. Library: $\log P(\texttt{True})$ at the library's position. Renormalised: `True` against `False` there. Labels in place: `(A` against `(B` there. Moved: `A` against `B` after the stem `The possible answer is (`. Deleted: Library without the quoted instruction.

	Library	Renormalised	Labels in place	Moved	Deleted
MMLU	0.503 [0.464, 0.540]	0.660 [0.624, 0.694]	0.563 [0.520, 0.606]	0.632 [0.595, 0.669]	0.587 [0.552, 0.623]
TriviaQA	0.473 [0.448, 0.497]	0.860 [0.836, 0.882]	0.857 [0.833, 0.880]	0.868 [0.844, 0.890]	0.731 [0.704, 0.758]

The library's P(True) confidence is indistinguishable from chance at ranking the model's correct answers on MMLU and below chance on TriviaQA; Qwen3-4B's library values are 0.423 $[0.369, 0.477]$ and 0.185 $[0.147, 0.226]$. After the stem, Moved's declared set holds the position: on the three Qwen3 MMLU runs, its medians exceed 0.999 and 398, 479 and 494 of 500 items reach the floor. Its paired gain over the library score is 0.129 $[0.078, 0.181]$ on MMLU, and on TriviaQA it is above all fourteen library estimators in the single-answer comparison. Moved minus Renormalised is -0.028 $[-0.058, +0.003]$ on MMLU and $+0.008$ $[-0.002, +0.018]$ on TriviaQA. Of Library, Renormalised and Moved, only Moved's declared set holds the scored position; in the forecast task the stem ranks correctness where renormalising at the original position stays indistinguishable from chance (Section 6). For Labels in place, an answer letter is usually the argmax on MMLU and a label on TriviaQA; the stem adds 0.069 $[0.042, 0.095]$ and 0.011 $[0.001, 0.021]$ to it. On four further checkpoints, Phi-3, Mistral, falcon and SmolLM2, `True` and `False` also hold under half the mass at the library's position on every item of both tasks. On Qwen2.5-72B-Instruct, their median mass there is 9.85×10^{-7} on MMLU and 4.27×10^{-6} on TriviaQA. With a non-gold letter printed as the possible answer, the MMLU argmax is the gold C or D on 102, 132 and 80 of 500 items on the Qwen3 checkpoints (Appendix E).

5 How often the readout fails across checkpoints

Table 3 compares seven checkpoints from five architecture families, using the same 64 items and fixed note as Section 4.

Six low-mass cases and one partial reproduction. All seven production-prompt medians meet the collapse definition. Six are at or below 5.8×10^{-4}, four of them below 10^{-5}, and every item on these six is below the floor, and no item's unrestricted argmax is a forecast option's first token. SmolLM2, the partial reproduction, has a median of 0.3895, with 50 of 64 items below the floor; 18 of its 64 argmaxes are an option's first token. Read the same way, the larger `Qwen3-32B` has a median of 3.79×10^{-3}; all 64 items are below the floor and none has a forecast option at the argmax. The stem lifts its median to 0.99928 (Appendix I). With Llama-3-8B-Instruct (median 2.565×10^{-4}, stemmed 0.99352), all nine checkpoints read with this prompt start below the floor and the stem lifts all nine above it. Table 3 gives each per-item minimum beside its median. The options' mass and the argmax check part on 13 of the 1024 item readings of Table 3 and Qwen3-32B, where the argmax is an option but the options hold under half the mass or the reverse (Appendix C.1).

Answer letters on five checkpoints, missing spaces on two. Answer letters occupy the scored position on most items for the three Qwen3 checkpoints, Phi-3 and SmolLM2 (Table 3). Mistral's most frequent next token is instead the bare space marker `'_'`, on 30 of 64 items. Answer letters take another 30, `'_D'` alone 27, a letter rate of 0.469. Falcon produces the space token `'Ġ'` on 35 of 64 and the newline `'Ċ'` on 29, with a letter rate of 0.000. Both rendered prompts omit final whitespace. Supplying the space puts a digit at the scored position and reduces Mistral's letter rate to 1 of 64; falcon's remains 0.000. On these two, at most one item in 64 keeps an answer letter once the separator is supplied (Appendix C.2).

Phi-3 has a second problem: its options share a first token even when encoded in context. Its median of 5.79×10^{-4} therefore measures option mass without establishing that the options are separable. Its answer-letter rate of 0.984 independently establishes the failure.

Table 3: Seven checkpoints on the same 64 frozen items, with one note held fixed: `Landlord only gives when told to take, not give.` It names no answer letter or option. Each checkpoint uses its own chat template, with `enable_thinking=False` where the template reads it; forward passes use float16 and log-softmax uses float32. Appendix C.1 gives the note source. μ (Appendix A) is the median over items, per-item minimum in parentheses; *Letter* is the fraction of items whose unrestricted argmax is an A–D answer letter; *Below* counts items under 0.5.

	Production prompt			With answer stem	
Checkpoint	μ (min)	Letter	Below	μ (min)	Below
Qwen3-4B-Instruct-2507	2.71e−15 (7.63e−19)	0.984	64/64	0.99940 (0.17702)	2/64
Qwen3-8B	8.14e−14 (6.97e−17)	0.953	64/64	0.99951 (0.92296)	0/64
Qwen3-14B	1.37e−8 (2.96e−14)	0.969	64/64	0.99999 (0.99966)	0/64
Phi-3-mini-4k-instruct	5.79e−4 (9.07e−5)	0.984	64/64	0.93344 (0.84818)	0/64
Mistral-7B-Instruct-v0.3	3.49e−6 (3.67e−8)	0.469	64/64	0.99050 (0.00216)	7/64
falcon-7b-instruct	1.05e−4 (5.70e−5)	0.000	64/64	0.53214 (0.33557)	5/64
SmolLM2-1.7B-Instruct	0.3895 (0.0352)	0.719	50/64	0.88834 (0.80527)	0/64

The answer stem's scope. The stem lifts all seven medians above the floor on Table 3's prompt. Falcon's argmax is a forecast option on every item, with winning-option probabilities between 0.122 and 0.315 and roughly 47% of mass off the grid. Mistral's worst item stays more than two orders of magnitude below the floor. The table gives per-item minima and below-floor counts.

Appendix I adds Llama-3-8B-Instruct. With per-item notes, a system-turn instruction, MMLU-Pro items (Wang et al., 2024c) or a sentence assigning the item's instruction to the readers, medians stay below the floor on 8, 6, 8 and 8 of eight checkpoints; an item answer letter is the unrestricted argmax on at least half the items, which we call *letter pull*, on 5, 5, 3 and 4 checkpoints, respectively (Table 15). Llama-3-8B-Instruct's article `A`, which opens prose such as `A clever question!`, is not counted as an answer letter. Adding a second sentence leaves five medians below the floor.

Two whole forecast templates, T1 and T2, written by a language model from a task description without knowledge of the failure, quote the same item block and end in an explicit output contract. Both remove letter pull on every checkpoint. Each leaves three medians below the floor, none of them from letter pull: Mistral's and falcon's, whose templates omit the final separator, with Phi-3's under T1, whose options share a first token, and SmolLM2's under T2, which puts an option at the argmax on 56 of 64 items with no item reaching the floor (median 0.31081). With their own answer stems, `Estimate:` for T1 and `Predicted number correct:` for T2, all eight medians clear the floor under T1 and seven under T2. Falcon's stemmed medians are 0.45421 with MMLU-Pro and 0.33207 under T2 (Tables 14 and 15).

6 Moving the scored position and checking the readout

The downstream forecast. The practical question is whether the scored forecast tracks the correctness of its note-only reader. Across 240 passages on each of the three Qwen3 checkpoints and Llama-3-8B-Instruct, Table 4 compares the original and stemmed positions. At the original position, the Qwen3-averaged AUC interval includes chance, and every Qwen3 passage is below the mass floor (Appendix H). The declared set and renormalisation are fixed and the stem moves

the scored position, so the paired Qwen3-averaged gain, 0.1223 $[0.0449, 0.1993]$, compares the two positions under one readout; Qwen3-14B's own difference interval includes zero. Both readouts use the same model, precision, GPU, note, item and option first-token IDs (Appendix H). On Llama-3-8B-Instruct, AUC rises from 0.5231 to 0.6044, a paired difference of 0.0813 $[0.0046, 0.1574]$. The stemmed Brier scores, 0.2544 for the Qwen3 average and 0.2522 for Llama-3-8B-Instruct, remain worse than the 0.2500 of a constant 0.5 forecast; Table 12 adds constants fitted to the evaluation rows. In Table 13's second reading of these passages, on other hardware, a short stem, `Number:` , gives a Qwen3-averaged AUC of 0.6145 against the task stem's 0.6156, a difference of -0.0011 $[-0.0213, 0.0191]$; for `Forecast:` the difference is -0.0084 $[-0.0328, 0.0162]$ (Appendix H). Deleting the item's instruction instead of adding the stem ranks reader correctness worse: averaged over the four checkpoints, its AUC is lower than the stem's by 0.0489 $[0.0191, 0.0791]$, although on the Qwen3 checkpoints the deletion also returns the options' median mass above 0.9998. On the same four-checkpoint average, template T1 (Section 5) comes within 0.0063 $[-0.0160, 0.0291]$ of the stem's AUC with its own answer stem; without it, the gap is 0.0250 $[-0.0019, 0.0518]$. Reading the number off the model's own unrestricted continuation of the production prompt ranks worse than the stem too, by 0.0772 $[0.0355, 0.1169]$ on that average, with a continuation lacking a grid number scored 0.5; those forecasts are nearly constant, 70 on 232 of Qwen3-14B's 240 passages (Appendix H).

Table 4: The forecast scored against the correctness it predicts, at the original and stemmed positions, on the same 240 passages at each of the three Qwen3 checkpoints and Meta-Llama-3-8B-Instruct. A Qwen3-averaged row is the mean of the three checkpoints' values; Llama-3-8B-Instruct is separate. Intervals are 95% percentile intervals of a paired cluster bootstrap over passages, $B = 20{,}000$, both readouts resampled on the same replicates.

	Original position	With answer stem	Difference
AUC			Gain, stemmed − original
Qwen3-averaged	0.4929 [0.4390, 0.5458]	0.6152 [0.5602, 0.6686]	0.1223 [0.0449, 0.1993]
Qwen3-4B	0.4958	0.6283	0.1325 [0.0168, 0.2459]
Qwen3-8B	0.4864	0.6435	0.1570 [0.0431, 0.2685]
Qwen3-14B	0.4964	0.5739	0.0775 [−0.0199, 0.1744]
Llama-3-8B-Instruct	0.5231 [0.4486, 0.5961]	0.6044 [0.5312, 0.6764]	0.0813 [0.0046, 0.1574]
Brier			Reduction, original − stemmed
Qwen3-averaged	0.4749	0.2544	0.2205 [0.1844, 0.2559]
Qwen3-4B	0.4487	0.2693	0.1794 [0.1399, 0.2190]
Qwen3-8B	0.5350	0.2463	0.2887 [0.2400, 0.3353]
Qwen3-14B	0.4409	0.2475	0.1933 [0.1439, 0.2414]
Llama-3-8B-Instruct	0.2640	0.2522	0.0118 [−0.0152, 0.0387]

What the answer stem changes. For Table 3's production prompt, the stem survives tokenisation and decoding at the end of the context on all seven checkpoints and lifts their medians above the floor. On five checkpoints the context ends in whitespace both before and after adding the stem, yet the medians rise from at most 0.390 to at least 0.888. Trailing whitespace alone therefore does not explain the increase. On Phi-3, the stem also restores distinct first tokens by ending the stemmed context in a space rather than a newline. Our *fence* instead wraps the quoted item in BEGIN/END markers and adds a request not to answer it, adapted from the DIRECT prompt of Hwang et al. (2025, Section 5.1), tested there on Llama-3.1-70B-Instruct translation. It leaves 43 of 64 Qwen3-4B items below the floor, compared with two for the stem (Appendix C.4).

First-token scoring needs distinct first tokens. Tokenising each option on its own gives every option the same initial word-boundary token on the two sentencepiece checkpoints among the seven, where the resulting mass measures that token's probability. Encoding each option as a continuation of the rendered prompt removes this isolated-encoding error, but the prompt's ending can still produce a shared first token. We cross three prompt forms (original, fenced item, original with the answer stem) with three item endings (original, instruction deleted, summarisation instruction). On Phi-3, the six combinations without the stem share a first token and the three stemmed ones have distinct first tokens; the other six checkpoints have distinct first tokens on all nine (Appendix C.1). Moving the scored position requires repeating this check.

A large mass cannot substitute for distinctness. G-Eval's prompt (Liu et al., 2023) ends at `- Coherence:` with no separator; adding a space to each G-Eval option raises the mass on Qwen3-4B from 1.26×10^{-5} to 0.99449, but all five options then share the space token on all seven checkpoints. The prefix trades the separator defect for the shared-first-token defect. Which defect a declared set meets is decided by how the vocabulary tokenises it, not by the strings alone (Appendix C.6).

Agreement with ratings. Appendix F gives the four specifications cell by cell (Liu et al., 2023; Zhuang et al., 2023; Dubois et al., 2024; Kadavath et al., 2022). On G-Eval's space-prefixed options, first-token scores are constant on all seven checkpoints, so their rank correlations are undefined. Supplying the separator in the context and scoring full option sequences makes the comparison defined, with correlation changes in both directions (Appendix G).

Checks before using a constrained-option score. Each check addresses a failure measured here.

1. Measure the options' in-context first-token mass at the scored position. A median below 0.5 means that on at least half the items most next-token probability lies outside the options' first tokens (Section 3).
2. Check the unrestricted argmax at the scored position. An answer letter where a forecast is requested flags a possible answer to the quoted item (Section 4).
3. Inspect the unrestricted continuation from the scored position. It shows whether the model answers the quoted item before forecasting, and so what role a flagged token plays (Figure 1).
4. Verify that the options' first tokens are distinct when encoded as continuations of the rendered prompt. Shared first tokens cannot distinguish the options, even at high mass (Section 6).
5. Inspect the rendered prompt's final characters and test a missing separator. A chat template that omits it can put whitespace at the scored position (Section 5).
6. Test an answer stem, repeat these checks after it and recheck the downstream target. On Table 3's prompt the stem lifts all seven median masses above the floor; the Brier score delimits what that lift establishes (Section 6).

7 Limitations and conclusion

The main mass and token-class comparisons use RACE-H, one forecast template, 64 items per checkpoint and one fixed note. Table 1 gives the other sample sizes; Appendix I varies the note, dataset, turn and template and adds Llama-3-8B-Instruct. The answer-stem result on nine checkpoints is scoped to Table 3's prompt. The Qwen3 checkpoints hold architecture and tokeniser fixed; from 4B to 32B, each keeps its production-prompt median below the floor, and the answer stem lifts each above it. Scale varies only within Qwen3, so these comparisons establish no size trend across families. SmolLM2 tests whether the failure is specific to Qwen3, not whether it is specific to small models.

Two observations remain unexplained. Deleting the imperative reduces Mistral's median option mass at the original ending from 3.49×10^{-6} to 1.77×10^{-6}, with the separator still omitted. The token-class rules also fail to predict which class replaces the answer letters. The method requires log-probabilities for chosen tokens at a chosen position, so these measurements do not address API-only models.

A constrained-option score reports a number even when the model is about to emit something else. Self-evaluation and judge scorers have this configuration when the quoted material carries its own answer instruction and the scorer reads a different declared set (Appendix E). Inspecting the unrestricted continuation identifies what is being answered; checking the tokenised options and rendered ending identifies what is being scored. The paired downstream comparison then tests whether that score tracks the intended target.

Reproducibility Statement

Section 3 and the appendices give the readouts, prompts, sampling and statistical comparisons. The supplementary material provides the per-item readings behind Table 3 and its Qwen3-32B extension, the forecast comparison of Table 4 and Appendix H, and the lm-polygraph comparisons of Section 4 and Appendix E, together with the human and judge labels behind the execution-label agreement reported in Section 4; its `README.md` names the fields and the resampling convention

for recomputing each statistic. Appendix A.2 pins the checkpoints of Table 3 and the checks that confirm them; Appendices H and I name the snapshots of the checkpoints they add; the cited `lm-evaluation-harness` counts and source locations use the revision pinned in Appendix A.2.

USE OF AI

In this work, we used generative AI tools to generate synthetic data sets, to search for, identify and summarise relevant literature, and to edit and polish the text of this paper for readability. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, to interpret results, to propose or refine hypotheses, to design or give feedback on research methodology or experiments, to implement methods, to clean and reformat datasets, or to support qualitative and thematic data analysis. Formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, and translation are not applicable to this work. Additionally, we used generative AI tools for the creation of artifacts.

Separately from assistance with the research, language models are components of the experiments, reported with their results: three judge models label whether each continuation executes an instruction (Section 4, Appendix D), with their labels checked against two independent three-person labellings, and a language model wrote the two forecast templates T1 and T2 from a task description, without knowledge of the failure, as test inputs (Appendix I). A language model also wrote the synthetic test inputs with known answers that check the labels: the twelve validation cases of Appendix D and the ten practice records of Appendix I.

We have reviewed all AI-assisted work. The papers we cite were read in full, and a citation was checked at the place in the cited paper that supports the claim it carries; the authors edited the text and verified it against the results it reports.

We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Silvia Cappelletti, Tobia Poppi, Samuele Poppi, Zheng Xin Yong, Diego Garcia-Olano, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Improving LLM first-token predictions in multiple-choice question answering via output prefilling, 2025. arXiv:2505.15323.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. Length-controlled AlpacaEval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir R. Radev. SummEval: Re-evaluating summarization evaluation, 2020. arXiv:2007.12626.

Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, Timothy Baldwin, and Artem Shelmanov. LM-Polygraph: Uncertainty estimation for language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 446–461. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-demo.41.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, et al. A framework for few-shot language model evaluation. `https://github.com/EleutherAI/lm-evaluation-harness`, commit `d6de816`, 2026.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, pp. 79–90. Association for Computing Machinery, 2023. doi: 10.1145/3605764.3623985.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. arXiv:2009.03300.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7038–7051, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.564.

Yerin Hwang, Yongil Kim, Jahyun Koo, Taegwan Kang, Hyunkyung Bae, and Kyomin Jung. LLMs can be easily confused by instructional distractions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19483–19496. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.957.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 1601–1611, 2017. arXiv:1705.03551.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Ahmed Karim, Fatima Sheaib, Zein Khamis, Maggie Chlon, Jad Awada, and Leon Chlon. Attention deficits in language models: Causal explanations for procedural hallucinations, 2026. arXiv:2602.19239.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2017.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using GPT-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023.

Joshua Robinson, Christopher Michael Rytting, and David Wingate. Leveraging large language models for multiple choice question answering, 2023. arXiv:2210.12353. Published at the Eleventh International Conference on Learning Representations, 2023.

Ananya B. Sai, Tanay Dixit, Dev Yashpal Sheth, Sreyas Mohan, and Mitesh M. Khapra. Perturbation CheckLists for evaluating NLG evaluation metrics. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021. arXiv:2109.05771.

Mario Sanz-Guerrero, Minh Duc Bui, and Katharina von der Wense. Mind the gap: A closer look at tokenization for multiple-choice question answering with LLMs, 2025. arXiv:2509.15020.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=RIu5lyNXjT.

Shreya Shankar, J.D. Zamfirescu-Pereira, Björn Hartmann, Aditya G. Parameswaran, and Ian Arawjo. Who validates the validators? Aligning LLM-assisted evaluation of LLM outputs with human preferences, 2024. arXiv:2404.12272.

Youran Sun, Xingyu Ren, Kejia Zhang, Xin-Peng Liu, and Jiaxuan Guo. PerspectiveGap: A benchmark for multi-agent orchestration prompting, 2026. URL https://arxiv.org/abs/2606.08878. arXiv:2606.08878.

Eric Wallace, Kai Xiao, Reimar Leike, Lilian Weng, Johannes Heidecke, and Alex Beutel. The instruction hierarchy: Training LLMs to prioritize privileged instructions, 2024. arXiv:2404.13208.

Xinpeng Wang, Chengzhi Hu, Bolei Ma, Paul Röttger, and Barbara Plank. Look at the text: Instruction-tuned language models are more robust multiple choice selectors than you think, 2024a. arXiv:2404.08382. Published at the First Conference on Language Modeling, 2024.

Xinpeng Wang, Bolei Ma, Chengzhi Hu, Leon Weber-Genzel, Paul Röttger, Frauke Kreuter, Dirk Hovy, and Barbara Plank. "my answer is c": First-token probabilities do not match text answers in instruction-tuned language models, 2024b. arXiv:2402.14499. Published in Findings of the Association for Computational Linguistics: ACL 2024.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2024c. arXiv:2406.01574.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12697–12706. PMLR, 2021. URL https://proceedings.mlr.press/v139/zhao21c.html.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors, 2023. URL https://arxiv.org/abs/2309.03882. Published at the Twelfth International Conference on Learning Representations, 2024.

Honglei Zhuang, Zhen Qin, Kai Hui, Junru Wu, Le Yan, Xuanhui Wang, and Michael Bendersky. Beyond yes and no: Improving zero-shot LLM rankers via scoring fine-grained relevance labels. *arXiv preprint arXiv:2310.14122*, 2023. Published at NAACL 2024.

Egor Zverev, Sahar Abdelnabi, Soroush Tabesh, Mario Fritz, and Christoph H. Lampert. Can LLMs separate instructions from data? And what do we even mean by that? In *The Thirteenth International Conference on Learning Representations*, 2025.

A Definitions and sampling

The original ending is the unmodified rendered prompt; the space-appended ending adds one space to it.

Section 3 defines μ in words; this is the closed form. Each option $o \in \mathcal{O}$ is tokenised *as a continuation of* the rendered context c: if $\mathrm{tok}(c)$ prefixes $\mathrm{tok}(c\|o)$, the option's tokens are the suffix and $t_1(o \mid c)$ its first. If this prefix condition fails for any option, the implementation marks the boundary as not preserved and records no in-context mass (`None`), rather than zero. When the condition holds for every option,

$$\mu(c) \;=\; \sum_{t \,\in\, \{\, t_1(o|c) \,:\, o \in \mathcal{O} \,\}} p(t \mid c). \tag{2}$$

When no option's token sequence is a prefix of another's, the full options' probabilities sum to at most $\mu(c)$. The sum counts each token once and measures separability only when the options' first tokens are distinct. On Qwen3-4B's 16 ladder items with the fence, the stem and space-prefixed options, a list-based sum reports a median of 4.9850 where the set sum is 0.997, and equal token lengths do not detect the duplicate first tokens. Scoring after a supplied shared prefix measures a different conditional position, giving Phi-3 a median of 0.0368 instead of 0.9334 on the 64-item stemmed prompt; Appendix C.5 gives the 16-item reading, 0.0392 against 0.9331.

A.1 Sampling

RACE-H is read from Hugging Face `ehovy/race`, configuration `high`, at revision `2fec9fd81f1dc971569a9b729c43f2f0e6436637`. We read `test`, `validation`, then `train` in stored row order, grouping rows by the MD5 of the UTF-8 passage text, keeping the first occurrence's position and accumulating its questions in encounter order. We keep the 19,566 passages of at least 200 words, shuffle them with `random.Random(0)`, and choose one question for each of the first 3,052 passages with the same generator. Every RACE-H sample in this paper is a prefix of this list, including the 64 items per checkpoint and the judge's first 32, shared by all 49 checkpoint–ending combinations.

A.2 Reproduction snapshots

Table 3 reads `Qwen/Qwen3-4B-Instruct-2507` at `cdbee75`, `Qwen/Qwen3-8B` at `b968826`, `Qwen/Qwen3-14B` at `40c0698`, `microsoft/Phi-3-mini-4k-instruct` at `f39ac1d`, `mistralai/Mistral-7B-Instruct-v0.3` at `c170c70`, `tiiuae/falcon-7b-instruct` at `8782b5c` and `HuggingFaceTB/SmolLM2-1.7B-Instruct` at `31b70e2`. Every weight and tokenizer file the runs read matches the Hub's record for its revision. Loaded at these revisions, Qwen3-8B, Qwen3-14B, Mistral and falcon reproduce their rows item for item on the GPU model they were read on; Qwen3-4B, Phi-3-mini and SmolLM2 were read on a second GPU model, and their re-read differs as Appendix I describes for C0. All cited lm-evaluation-harness source locations and task counts use commit `d6de81643928d653435c431bae19945d41d32520`.

B The prompt, rendered

This is the complete forecast prompt for `high6240.txt` on `Qwen/Qwen3-4B-Instruct-2507`, the example in Figure 1. Its last 60 characters, its 178 tokens and its last token id match the run that scored it. The constructed note contains the passage's first two sentences and its gold answer. The checkpoint's chat template opens the assistant turn below.

```
<|im_start|>user
Note:
A woman from Japan was telling a friend about her trip to the United
↪  States. The woman had visited major businesses and investment
↪  companies in New York City and Chicago. The correct answer is A: her
↪  English was poor.

A future reader will be shown ONLY the note above, not the source text,
↪  and will then be asked:
After the woman visited the United States she might feel that   _  .
A: her English was poor
B: it's easy to master English
C: it's difficult to make money
D: people there weren't very friendly
Answer with exactly one of: A, B, C, D.

Out of 100 such readers, how many answer correctly? Reply with exactly
↪  one of: 10, 30, 50, 70, 90.<|im_end|>
<|im_start|>assistant
```

The rendered string has 704 characters; its SHA256, computed over UTF-8 with no added trailing newline, is `7db81d9513ef11e6c7bd8f73ad7a8459cb4b04dac22d987419f16db9dc7246d9`. Arrows mark typeset line breaks. The string ends with the newline after `assistant`; the forecast instruction follows the item's letter instruction.

The scored position. The readout scores the next token after zero-based index 177, the newline (id `198`) closing the assistant opening. The five options are `10`, `30`, `50`, `70`, `90`; each has two tokens with distinct first-token ids `16`, `18`, `20`, `22`, `24`.

On the 60 rows of the 120-row probe (Appendix C.3) whose notes concern an unrelated passage and so state no answer, an A–D letter is still the unrestricted argmax on 58. The forecast argmax and the reader task's argmax are the same token on 59 of the 60 rows whose notes state the answer, against 38 of the 60 whose notes do not.

The scored position with the stem. Appending `"Out of 100 such readers, about "` to the rendered string, after the assistant opening, adds 11 tokens; scoring follows index 188, the trailing space (id `220`). On this item, the argmax is `'5'` at 0.98907 and the declared set holds 0.9999999 of the mass; the unrestricted continuation is `"50 answer correctly.\n\n50"`.

C SUPPORTING MEASUREMENTS

C.1 CROSS-FAMILY READINGS

The fixed note. Table 3 uses Qwen3-4B's note for item 58 (`high2565.txt`), with its quotation marks removed. Its full text is printed in the table caption. With the answer stem, falcon's argmax is `'90'` on 49 of 64 items and `'10'` on the other 15; its winning probabilities range from 0.1225 to 0.3147, with median 0.2133.

Option mass and the argmax check. On Table 3's two prompts and the same prompts for Qwen3-32B, 1024 item readings in all, the options' mass and the argmax check disagree on 13. SmolLM2's production prompt puts an option at the argmax on 6 items whose options hold under half the mass and an answer letter there on 2 items whose options hold more than half; falcon's stemmed prompt puts an option at the argmax on 5 items under half. On the other 14 checkpoint and prompt pairs the two agree on every item.

The nine combinations cross three prompts (production, fenced, and production with the answer stem) with three quoted-item endings (the letter instruction, its deletion, and its replacement by `Summarise in one sentence.`). Across these combinations, Phi-3's six unstemmed combinations share a first token and its three stemmed combinations have distinct first tokens; the other six checkpoints have distinct first tokens on all nine. Isolated and in-context masses coincide bitwise on all nine combinations for the five checkpoints with distinct isolated first tokens. The two sentencepiece cases are compared in Appendix C.5.

C.2 DETAILS OF ANSWER-STEM AND BOUNDARY CHECKS

The boundary rule's two exposure classes require opposite values of whether the rendered context ends in whitespace (Table 5). Exposure is checked against in-context token distinctness and the unrestricted argmax. The stem restores distinct first tokens on Phi-3 and supplies the missing separator on Mistral and falcon; it leaves the whitespace condition unchanged on five checkpoints whose medians rise from at most 0.390 to at least 0.888. Qwen3-4B's unstemmed forecast has neither boundary defect and still begins with an answer letter on 0.984 of items. These checks depend on the rendered context, so the variant has to be named on every reading.

Table 5: Boundary exposure classes for the measured checkpoints and tokeniser versions. The rule separates endings with and without whitespace; the text reports which defects occur.

Prompt ends with	Vocabulary	Consequence	Checkpoints
Newline	sentencepiece	marker restored, mass not a reading	Phi-3-mini
Newline	byte-level	neither problem	Qwen3 $\times$3, SmolLM2
`[/INST]`, `Assistant:`	either	separator omitted by template	Mistral, falcon

The paired separator controls. Two checkpoints have the omitted-separator defect. Neither has an answer letter as its modal token across the items of any reading in Sections 4 to 6; in Appendix I, under the system-turn condition S1, Mistral has a letter at the argmax on 57 of 64 items (Table 14), with `A` the modal token on 18. `Mistral-7B-Instruct-v0.3` has an answer-letter rate of 0.469 at the original ending, on 30 of 64 items, histogram 30 whitespace tokens, 30 answer letters and 4 prose words. Deleting the item's instruction reduces the rate to 0.016, leaving 63 whitespace argmaxes and one answer letter. Supplying the separator the template omits, with the instruction kept, takes the rate to 1 of 64, with 63 digits and 1 answer letter: 29 of the 30 letter items disappear. Both interventions remove all but one of Mistral's letters: at the original ending they depend on the instruction, and supplying the separator removes them even with the instruction intact. With the separator supplied, 56 of the digits are grid-option first tokens and seven are `0`. Falcon shows the same defect with 0 of 64 either way, which is why the pair is needed to tell the two apart; supplying the separator puts a grid option's first token at the argmax on 63 of 64 falcon items. For substitution, both checkpoints use the space-appended ending; the deletion comparison above uses the original ending.

C.3 SCORED-POSITION PROBE ON 120 ROWS (SECTION 4)

The 120 RACE-H rows use 60 notes containing the passage's opening text and gold answer, and 60 notes about an unrelated passage. An A–D answer letter is the unrestricted argmax on 118 of these rows, and no forecast option is the argmax. The median option mass is 1.42×10^{-15}; its extremes are 1.73×10^{-19} and 4.64×10^{-6}. The two letter exceptions are `high1845.txt`, whose argmax is `'None'` at 0.868, and `high14114.txt`, whose argmax is `'0'` at 0.99998.

C.4 PROMPT LADDER

The request not to answer the quoted item adapts the DIRECT prompt of Hwang et al. (2025, Section 5.1, Table 4), tested on Llama-3.1-70B-Instruct for translation; the BEGIN/END markers are ours. On Qwen3-4B's 64 fixed-note items, quoting the item gives a median mass of 2.71×10^{-15} and 0 of 64 option argmaxes; fencing gives 3.74×10^{-3} and 21 of 64, with mass from 9.10×10^{-9} to 0.99995; the stem gives 0.99940 and 62 of 64. Below-floor counts are 43 with the fence and two with the stem.

C.5 TWO MISLEADING READINGS FROM ISOLATED ENCODING

The columns below compare isolated, divergent (scored after the options' shared prefix; Appendix A) and in-context readings on the same forward passes, for the two cases below.

For the first, on `Phi-3-mini-4k-instruct`'s prompt with the answer stem, the isolated column reads 1.27×10^{-8} and the divergent companion 0.0392 with 0 of 16 argmax hits, while Eq. 2 on those same 16-item passes gives 0.9331, minimum 0.8977, and 16 of 16 hits. The argmax there is `'7'` in 9 items, `'3'` in 6 and `'5'` in 1: the model emits a grid digit while the isolated estimator scores a boundary marker after a space. For the second, `Mistral-7B-Instruct-v0.3` reads 0.4545 in isolation, which appears to be substantial option mass but is $p(_)$. Eq. 2 gives 3.93×10^{-6} with 0 of 16 hits on the original ending, where the template omits the separator.

Table 6: Isolated and divergent columns beside the in-context reading, all from the same 16-item forward passes. Table 3 gives the 64-item medians. A separate 16-item reading without the in-context column has identical isolated and divergent values to every digit.

			Alternative columns		Eq. 2
Reading	Checkpoint	Prompt	isolated	divergent	in-context
Stem repairs the readout	Phi-3-mini-4k-instruct	stemmed	1.27e−8	0.0392	**0.9331**
Original-ending option mass	Mistral-7B-Instruct-v0.3	unstemmed	0.4545	0.9935	3.93e−6

C.6 DECLARED SETS AND VOCABULARY

Across three declared sets and 11 checkpoints using seven vocabularies, {`' (A)'`, `' (B)'`} shares a first token on all 11 and {`' no'`, `' yes'`} stays distinct on all 11. The spaced grid {`' 1'`,...,`' 5'`} shares a first token on the seven main checkpoints and stays distinct on GPT-2 and the three Flan checkpoints, whose vocabularies merge the separator with the option's first characters. A rule that predicts distinct first tokens whenever the characters after the separator differ therefore gives seven false clearances and no false alarms across these 33 cells. The Qwen3 models share their base vocabulary, and Flan-T5-XL, Flan-T5-XXL and Flan-UL2 share one SentencePiece model.

C.7 THE READER TASK WITH SPACED AND UNSPACED OPTIONS

The spaced option strings' median mass is 5.21×10^{-14} where the note contains the answer and 9.67×10^{-12} where it does not. The unspaced strings take the median to 1.000 and the argmax-on-option rate to 1.00 where the note carries the answer and 0.967 where it does not, while 96.25% of picks are unchanged and overall accuracy moves from 0.6958 to 0.7208. That is the same encoding defect as Section 6, in a third place. The spaced strings' two group accuracies are 55.8 points apart on a four-way item, read out of the tail with the argmax on an option in none of the 240 rows.

Table 7: The reader task with spaced and unspaced options, same 240 rows, same run. Mass is the median option first-token mass by group; accuracy is against the passage's gold answer. The spaced and unspaced strings select the same option in 96.25% of rows.

	Mass (median)		Argmax on option		Accuracy	
Option strings	gold in note	unrelated	gold in note	unrelated	gold in note	unrelated
`" A"–" D"`	5.21e−14	9.67e−12	0.00	0.00	0.975	0.4167
`"A"–"D"`	1.000	0.99999976	1.00	0.967	0.9833	0.4583

On the passages of Table 4. Each Qwen3 checkpoint's note-only reader is read again on its own notes: with the spaced strings, which reproduces every stored pick; with the unspaced strings; and by a greedy continuation of at most 16 tokens, taking its first standalone letter. The spaced and unspaced picks agree on 232, 231 and 236 of 240 rows on Qwen3-4B, Qwen3-8B and Qwen3-14B, and the unspaced pick equals the generated letter on every row whose continuation names one (240, 239 and 240 rows). Scored against the unspaced reader, the forecast's AUC moves from 0.4983, 0.4884 and 0.5019 at the original position to 0.6190, 0.6419 and 0.5799 with the stem, and the Qwen3-averaged AUC from 0.4962 $[0.4421, 0.5494]$ to 0.6136 $[0.5568, 0.6688]$, a paired gain of 0.1174 $[0.0382, 0.1969]$. Against the generated answers, the averaged AUC moves from 0.4968 to 0.6132, a gain of 0.1163 $[0.0371, 0.1956]$. At both positions the Brier scores move by less than 0.01. Intervals use Table 4's passage bootstrap.

D THE INSTRUCTION EXPERIMENT

The instruction comparison uses 64 items per checkpoint with the note held fixed. The token-class readings and the judge's rows use the original ending on five checkpoints and an appended space on Mistral and falcon; the deletion comparisons of mass and of Section 4's letter rates use the original ending on all seven (Appendix C.2). The six substitutes for the item's own `Answer with exactly one of: A, B, C, D.` are the empty ending, `Summarise in one sentence.`, `Reply with a single integer.`, `List three keywords from the passage.`, `Translate the passage into French.` and `Count the sentences in the passage.` The list, translation and counting instructions name `the passage`; the prompt gives the model the note and quoted item.

Deleting the instruction and the choice list. Deleting only the lettered choice list leaves answer-letter rates of 0.891, 0.781, 0.672 and 0.719 on Qwen3-4B, Qwen3-8B, Qwen3-14B and Phi-3, respectively; SmolLM2 rises from 0.719 to 0.828. With the instruction already deleted, Phi-3 reads 0.500 with the choices and 0.000 without them; the other six read at most 0.016 either way. With the original rendered ending, deleting the multiple-choice imperative raises the median mass by factors of 3.7×10^{14}, 1.2×10^{13} and 7.3×10^{7} on the three Qwen3 checkpoints in Table 3 order, 1.49 on SmolLM2 and 1.39 on falcon; it falls by a factor of 0.507 on Mistral. The last two readings are at the omitted separator, where deleting the imperative leaves the template defect. Deletion raises the median above the floor on four checkpoints: the three Qwen3 checkpoints and SmolLM2, which rises from 0.3895 to 0.58023; falcon begins at 1.05×10^{-4} and stays below the floor. Phi-3 has shared first tokens both with and without the imperative, so only its letter rate is interpretable.

Token-level rules. The opening-word rule asks whether the argmax is the substituted instruction's own opening word; over the list, translation and count instructions on the seven checkpoints, its sole nonzero rate is Qwen3-14B under the list instruction, 0.015625, and it reads 0.0000 on all seven translation cells. On `Qwen3-4B-Instruct-2507` the argmax is instead the opening of a response that carries the instruction out, `'Keywords'` in 17 of 64 items under the list instruction and `'Note'` in 30 of 64 under translation, so an opening-word match and execution of the instruction measure different properties. The token-class rule predicts an answer letter for the item's own instruction, a digit for the empty ending and the numeric instruction and a prose word for the other four; a check holds when the predicted class is the unrestricted argmax on more than half the items, and counted in the order of Table 3, the numbers of failing token-class checks are 1, 5, 3, 2, 5, 5 and 2.

The judges and the hand labels. Each judge answers every record with its id and one of the four label names, so a letter that answers the quoted item cannot be read as a label; for a substituted instruction, the rubric ranks execution above either answer label and leaves only answer versus forecast unranked. Grok, Claude Opus 5.5 and GPT-6-astra label the same 1862 records. Each matches the assigned label on all twelve synthetic cases, six executing and six not. Each labels non-executing all 126 cross-paired rows, which are judged against an instruction they were not given, and labels executing all 13 real rows that a separate first-token heuristic selects, 11 of them from `Qwen3-4B-Instruct-2507`. The heuristic accepts translation openers `Le, La, Les, L, Voici, Il, Elle, Dans, Traduction, Un, Une, Ce, Cette, Lorsque, Quand, Selon` or list headings `Keywords, keywords, Mots, Mot`, after stripping leading token-boundary markers, and rejects digit-opening translation and list rows. Both instruments use the same first 32 items in the same order, prompt and ending variant: the continuation begins with the token probe's argmax on 1566 of 1568 rows, with both exceptions on SmolLM2. Execution rates count any continuation consisting only of a number as a forecast, regardless of its judge label, which changes no translation, listing or summarisation cell; a bare number can satisfy both `Reply with a single integer.` and the forecast question, so the numeric instruction has no attributable execution rate. Three people labelled 112 continuations: 100 drawn from the experimental rows, covering the seven checkpoints and the seven endings unevenly (32 list and 25 translate rows, 8 or 9 for each other ending; 7 to 34 rows per checkpoint), and the twelve synthetic cases, which they were given first and not told apart from the rest. They saw the note, the quoted item, the ending and the continuation, and not the judge's label. A second, independent three-person labelling covers the same 100 continuations. Every agreement, like the rates, counts a bare number as a forecast (Table 8).

Execution rates, item answers and forecasts. Each rate uses 32 items and 64-token greedy continuations. Grok's translation-execution rates are 0.6562, 0.7500, 0.9688, 0.8438, 0.2188 and 0.4375 on the six checkpoints other than falcon, in Table 3 order, and falcon reads 0.0000; Opus and GPT also read translation execution on those six and 0.0000 on falcon. The listing instruction reads 0.9062 on `Phi-3-mini-4k-instruct` and at most 0.3750 elsewhere. Summarisation reads at most 0.1875, on `Qwen3-8B`, and counting at most 0.5938, on `Qwen3-14B`. With the item's instruction intact, Grok labels 31, 32, 31, 32, 3, 0 and 21 of 32 continuations as item answers, in Table 3 order, Mistral and falcon at the space-appended ending; with it deleted, it labels 32, 32, 32, 18, 32, 31 and 31 as forecasts. Opus and GPT assign the same labels as Grok on every Qwen3 row in these two conditions and label all 32 Phi-3 base continuations as item answers.

Table 8: Agreement on execution labels, on the binary executes-or-not reading except in the last row, which compares the four-way label. Judge pairs are compared on the 1568 experimental continuations. Each labelling covers 100 continuations; comparisons with a judge or between labellings use its majority of three, and the pairwise row and Fleiss' κ use the individual labels.

Comparison	Rows	Agreement	κ
Grok and Opus	1568	1549	0.958 (Cohen)
Grok and GPT	1568	1532	0.922 (Cohen)
Opus and GPT	1568	1533	0.924 (Cohen)
First majority and Grok, Opus, GPT	100	99, 98, 99	
Second majority and Grok, Opus, GPT	100	100, 99, 100	
First and second majorities	100	99	
First labelling's three labellers, pairwise	100	99, 99, 100	0.983 (Fleiss)
Second labelling's three labellers	100		0.983 (Fleiss)
First majority and Grok, four-way	100	all but 6	

Agreement. Grok matches the first majority on all 99 rows the three label unanimously; the one row it misses is a majority of two on `translate`. It agrees with the human majority on all 45 digit-opening rows in the sample that the first-token heuristic rejects, including the 19 labelled executing. Both validation sets give identical executes-or-not labels at 64 and at 128 generated tokens (256 on the 36 cross-paired Mistral and falcon rows), and at 128 tokens translation execution rises on three of the five checkpoints using the original ending and is unchanged on two. Judged again in reshuffled batches, the 448 unchanged continuations of the two checkpoints using the space-appended ending

keep their binary execution labels on all 192 translation, listing and summarisation rows and on 426 of 448 overall.

E WHAT OCCUPIES A PUBLISHED SELF-EVALUATION READOUT

A maintained implementation. A P(True)-style self-evaluation has this configuration when the quoted item carries its own answer instruction and the scorer reads a different declared set. lm-polygraph (Fadeeva et al., 2023), a maintained library and benchmark for uncertainty estimation, implements P(True) among its estimators (their Table 1) and, at revision `32cdf4a7`, enables it in 23 of the 30 evaluation configurations it ships. On MMLU in its `simple_instruct` configuration, the quoted input opens with the instruction to reply with the selected option's letter only, lists the lettered options and ends with `Answer:`. The estimator appends the model's own answer after `Possible answer:`, asks whether it is `(A) True` or `(B) False`, ends with `The possible answer is:`, and reads the probability of the token `True` at the start of the assistant turn. We ran it with the library's own loaders, prompt and answer step on 64 items drawn by its subsampling (seed 1), in bfloat16 as the library loads, and reproduced its reported P(True) to a log-space difference of 0 on the Qwen3 checkpoints and 9.5×10^{-7} on Llama-3-8B-Instruct. As the library renders the hybrid Qwen3-8B and Qwen3-14B, leaving thinking on, `<think>` is the argmax on all 64. The library's answer step samples at each checkpoint's own generation settings and requires two new tokens, so printed answers include strings such as `B<|im_start|>`.

Correctness ranking on MMLU. Using the library setup above on 500 MMLU items, we label each generated answer by its first standalone A–D letter against gold; an answer without such a letter is incorrect. Table 9 compares four readouts for the same generated answer. Library is $\log P(\texttt{True})$, the negative of the PTrue estimator's output. Renormalised is $P(\texttt{True})/[P(\texttt{True}) + P(\texttt{False})]$ at the same scored position. Moved prefills the assistant turn with the answer stem `The possible answer is (` and reads $P(\texttt{A})/[P(\texttt{A}) + P(\texttt{B})]$, where `A` denotes True. Deleted removes the quoted letter instruction before applying the library readout. The Qwen3 average weights the three checkpoints without thinking equally. The answer stem raises the Qwen3-averaged AUROC to 0.632 $[0.595, 0.669]$; comparing answers within each answer letter, with letter-specific AUROCs weighted by the number of correct–incorrect pairs, gives a gain of 0.133 $[0.079, 0.187]$. Deleting the letter instruction raises the library readout's Qwen3-averaged AUROC by 0.085 $[0.042, 0.127]$. In the metric lm-polygraph uses to compare estimators, the prediction–rejection ratio (PRR; 0 for random rejection, 1 for the oracle) (Fadeeva et al., 2023), the library readout has Qwen3-averaged PRR 0.016 $[-0.092, 0.124]$, and -0.187 $[-0.353, -0.013]$ on Qwen3-4B, below random rejection; the answer stem raises the Qwen3-averaged PRR to 0.407 $[0.320, 0.486]$, a paired gain of 0.390 $[0.253, 0.524]$. With thinking on, as the library renders Qwen3-8B and Qwen3-14B, the answer step returns only `<think>` and a newline on all 500 items of each, so the estimator scores a thinking token and no answer is correct.

Mass and split at the library's scored position. Write $M = P(\texttt{True}) + P(\texttt{False})$ and $R = P(\texttt{True})/M$, so $\log P(\texttt{True}) = \log M + \log R$. On Qwen3-4B, correct answers tend to come with a smaller M, AUROC 0.353 $[0.302, 0.405]$, and, on the 387 items where an A–D letter is the most probable token at that position, with a more probable letter, AUROC 0.638 $[0.573, 0.700]$. R alone ranks correctness at 0.674, but Library carries the $\log M$ term and ranks correct answers below incorrect ones. The direction of M differs between checkpoints: on Llama-3-8B-Instruct its AUROC is 0.617 $[0.565, 0.668]$, and Library ranks above chance.

The floor and the ranking. At the library's scored position, the mass of the declared set `True`, `False` is below the floor on every item on the three Qwen3 checkpoints without thinking; of the four readouts, only Moved clears the floor at the median on all three. At that position an A–D letter is the unrestricted argmax on 387, 496 and 500 of 500 MMLU items on Qwen3-4B-Instruct-2507, Qwen3-8B and Qwen3-14B without thinking, respectively. Of these, 179, 215 and 129 are C or D, which the outer `(A)`/`(B)` labels do not offer. With the quoted letter instruction deleted, a letter is the argmax on 0, 0 and 3 items. Moved reads the library's own `(A)`/`(B)` labels after the stem: its declared set holds the scored position, with median mass above 0.999 and 398, 479 and 494 of 500 items at or above the floor on the three Qwen3 runs.

Renormalised and Moved both rank correctness on MMLU and TriviaQA (Tables 9 and 10). Section 4 reports the paired differences between these readouts. In the forecast task, renormalisation at the original position stays indistinguishable from chance, 0.4929 $[0.4390, 0.5458]$, while the stem reaches 0.6152 $[0.5602, 0.6686]$ (Table 4); Section 6 gives the paired comparison of the two positions.

Counting as correct the 11 Qwen3-4B answers with no standalone letter but a gold first letter fused to lowercase text, such as `Bropical`, gives a Qwen3-averaged gain of 0.120 $[0.067, 0.173]$.

A wrong printed answer. On the 500 MMLU items of Table 9, we replace lm-polygraph's printed possible answer with a non-gold letter drawn per item with a fixed seed. On Qwen3-4B-Instruct-2507, Qwen3-8B and Qwen3-14B, with thinking disabled on the latter two, a bare A–D letter is the unrestricted argmax at the library's scored position on 259, 500 and 500 items, respectively. Among these letter argmaxes, 186, 289 and 233 match the gold letter, 39, 105 and 117 match the printed letter, and 34, 106 and 150 match neither. Across all 500 items per checkpoint, the gold letter's median probability is 0.0626, 0.951 and 0.246, against 4.10×10^{-7}, 6.90×10^{-6} and 1.01×10^{-5} for the printed letter. On 102, 132 and 80 items, respectively, the unrestricted argmax is the gold `C` or `D`: it correctly answers the quoted item, and is neither the printed answer nor one of the outer `(A)`/`(B)` labels.

Table 9: lm-polygraph confidence on 500 shared MMLU items: AUROC for the checkpoint's own answer being correct under the letter label. Intervals are 95% item-bootstrap intervals ($B = 10{,}000$), with the same resampled items across readouts and checkpoints. The readouts are defined above; median mass is the declared set's at its scored position, shown as Library / Moved. The Qwen3 average weights checkpoint accuracies and AUROCs equally. Differences use unrounded AUROCs.

Checkpoint	Accuracy	Median mass Library / Moved	Library	Renorm.	Moved	Deleted	Moved − Library
Qwen3-4B-		1.8×10^{-11}	0.423				+0.187
Instruct-2507	0.724	1.00	$[0.369, 0.477]$	0.674	0.610	0.560	$[+0.113, +0.260]$
Qwen3-8B,		3.4×10^{-13}	0.474				+0.143
no thinking	0.764	1.00	$[0.413, 0.536]$	0.592	0.617	0.580	$[+0.057, +0.224]$
Qwen3-14B,		1.2×10^{-10}	0.611				+0.059
no thinking	0.798	1.00	$[0.550, 0.670]$	0.715	0.669	0.623	$[-0.017, +0.134]$
			0.503				+0.129
Qwen3 average	0.762	n/a	$[0.464, 0.540]$	0.660	0.632	0.587	$[+0.078, +0.181]$
Llama-3-8B-		3.3×10^{-4}	0.659				−0.061
Instruct	0.614	0.999	$[0.609, 0.708]$	0.618	0.597	0.607	$[-0.127, +0.005]$

A second task: TriviaQA. We use the library's loader, subsample, answer step and P(True) prompt on 500 TriviaQA items. An answer is correct when its first line, stripped of a leading `Answer:`, matches a gold alias after both are stripped of non-ASCII symbols and nonspacing marks and passed through the library's normalisation. The library's label uses the full answer without these extra removals: 257 of Qwen3-8B's answers start with `Answer:`, and 20 of Qwen3-4B's answers end in a non-ASCII symbol. At the scored position, one of the library's option-label tokens `(A` and `(B` is the unrestricted argmax on 500, 500 and 463 items on the three Qwen3 checkpoints without thinking and 396 on Llama-3-8B-Instruct. On those items the model writes the library's option label where the library reads `True`. Table 10 gives the masses and AUROCs. In PRR, Library scores -0.013 $[-0.071, 0.039]$ on the Qwen3 average and -0.506 $[-0.585, -0.433]$ on Qwen3-4B, and Moved scores 0.756 $[0.700, 0.805]$ on the average. With the library's own correctness label, the Qwen3-averaged AUROCs are 0.418 for Library and 0.866 for Moved. With thinking on, the answer step returns only `<think>` and a newline on all 500 items of each of Qwen3-8B and Qwen3-14B.

Leaving out one checkpoint. Dropping any one of the three Qwen3 checkpoints from the average keeps Moved's gain over the library readout above zero: from $+0.1007$ $[+0.0434, +0.1578]$ to $+0.1648$ $[+0.1024, +0.2266]$ on MMLU and from $+0.2424$ $[+0.2065, +0.2771]$ to $+0.5846$ $[+0.5351, +0.6332]$ on TriviaQA.

The labels at the library's position. Reading the prompt's own labels where the library reads `True`, without a stem, renormalises the probabilities of `(A` and `(B`, each one token on these four checkpoints.

Table 10: lm-polygraph confidence on 500 shared TriviaQA items: AUROC for the checkpoint's own answer being correct under the alias-match label defined above. Intervals are 95% item-bootstrap intervals ($B = 10{,}000$), with the same resampled items across readouts and checkpoints. The readouts are defined in the MMLU comparison; here Deleted removes `Answer the following question as briefly as possible.` from the quoted input. Median mass is the declared set's at its scored position, shown as Library / Moved. The Qwen3 average weights checkpoint accuracies and AUROCs equally. Differences use unrounded AUROCs.

Checkpoint	Accuracy	Median mass Library / Moved	Library	Renorm.	Moved	Deleted	Moved − Library
Qwen3-4B-Instruct-2507	0.344	6.4×10^{-15} / 1.00	0.185 [0.147, 0.226]	0.859	0.884	0.742	+0.699 [+0.636, +0.760]
Qwen3-8B, no thinking	0.450	7.4×10^{-12} / 1.00	0.395 [0.346, 0.446]	0.868	0.865	0.689	+0.470 [+0.403, +0.535]
Qwen3-14B, no thinking	0.518	5.1×10^{-8} / 1.00	0.839 [0.803, 0.872]	0.852	0.853	0.763	+0.015 [−0.005, +0.035]
Qwen3 average	0.437	n/a	0.473 [0.448, 0.497]	0.860	0.868	0.731	+0.395 [+0.360, +0.428]
Llama-3-8B-Instruct	0.604	2.0×10^{-6} / 1.00	0.664 [0.616, 0.712]	0.832	0.829	0.669	+0.164 [+0.116, +0.214]

On MMLU, where an answer letter is usually the argmax at that position, the labels' median mass on the three Qwen3 checkpoints is 2.5×10^{-7}, 2.5×10^{-10} and 2.4×10^{-12} (4B, 8B and 14B), and this readout's AUROC is 0.626, 0.561 and 0.502, averaging 0.563 $[0.520, 0.606]$. Moved minus this readout is -0.016 $[-0.048, +0.016]$, $+0.056$ $[+0.024, +0.088]$ and $+0.167$ $[+0.111, +0.223]$, and $+0.069$ $[+0.042, +0.095]$ on the average. On TriviaQA, where a label is usually the argmax and the labels' median mass is at least 0.981 on each Qwen3 checkpoint, the readout averages 0.857 $[0.833, 0.880]$ and Moved minus it is $+0.011$ $[+0.001, +0.021]$. On Llama-3-8B-Instruct, Moved minus it is $+0.037$ $[+0.002, +0.071]$ on MMLU and -0.004 $[-0.023, +0.015]$ on TriviaQA. The same runs reproduce the stored library readout, Renormalised and Moved on every item.

Four further checkpoints. The same test, on the same 500 items per task, runs on `Phi-3-mini-4k-instruct`, `Mistral-7B-Instruct-v0.3`, `falcon-7b-instruct` and `SmolLM2-1.7B-Instruct`. Mistral and SmolLM2 load through the library's loader; Phi-3 and falcon load through the native `transformers` classes, because their bundled modelling code does not run under the pinned `transformers` 5.17.0. The library-readout check holds exactly on every item. Across all eight checkpoints, `True` and `False` hold less than half the mass at the library's scored position on every TriviaQA item and on every MMLU item except one of Llama-3-8B-Instruct's; their largest median mass, SmolLM2's, is 0.023 on MMLU and 0.153 on TriviaQA. After the answer stem, the library's `(A)` and `(B)` labels hold at least half the mass on every TriviaQA item on all eight.

A 72B checkpoint. The same test runs on `Qwen/Qwen2.5-72B-Instruct` at `495f393`, through the library's loader in bfloat16, and the library-readout check holds exactly on every item. `True` and `False` hold a median mass of 9.85×10^{-7} on MMLU and 4.27×10^{-6} on TriviaQA, with item maxima of 1.5×10^{-3} and 6.7×10^{-3}. After the answer stem, the smallest mass the `(A)` and `(B)` labels hold on any item of either task is 0.847.

Single-answer estimator comparison on TriviaQA. Our comparison includes every entry in lm-polygraph's `default_estimators.yaml` at revision `32cdf4a7` that uses one generated answer without additional answer samples, auxiliary models or external data: fourteen library estimators including PTrue (Library). The other entries are `MaximumSequenceProbability`, `Perplexity`, `MeanTokenEntropy`, `MeanPointwiseMutualInformation`, `MeanConditionalPointwiseMutualInformation`, `SelfCertainty`, `RenyiNeg`, `FisherRao`, `AttentionScore`, `RAUQ` at its two configured settings, `CSL` and `BoostedProbSequence`. The additional estimators use their configured settings and the library's own calculators on the recorded answer tokens, with confidence the negative of uncertainty. On the first five items per Qwen3 checkpoint, the replay matches direct library generation in tokens and log-likelihoods; the attention replay also matches generation attention and both RAUQ values. On the Qwen3 average with the alias-match labels of

Table 10, Moved has the highest AUROC point estimate among the seventeen readouts: the fourteen library estimators plus Renormalised, Moved and Deleted. Library ranks twelfth of the fourteen. The highest library AUROC is 0.815 (`MaximumSequenceProbability`); Moved's paired gain over it is $+0.053$ $[+0.024, +0.082]$. Renormalised is second, at 0.860, with a paired gain over it of $+0.045$ $[+0.016, +0.074]$. Moved also has higher PRR than every library estimator at both default maximum rejection fractions. Its paired gain over the best library estimator is $+0.083$ $[+0.019, +0.148]$ at 1.0 (`MaximumSequenceProbability`) and $+0.100$ $[+0.028, +0.174]$ at 0.5 (the entropy variant of `RAUQ`). These are paired 95% item-bootstrap intervals using the same resampled items across readouts and checkpoints.

Where the configuration occurs. Self-evaluation and judging quote by design: P(True) quotes the question it asks about (Kadavath et al., 2022), and a log-probability judge quotes the instruction and the outputs it grades. The public pipelines below ship this configuration in their scorers when the quoted material carries its own answer instruction and the scorer reads a different declared set. lm-polygraph enables P(True) in 23 of its 30 configurations, and on MMLU the quoted input carries the instruction to answer with a letter. AlpacaEval, at revision `cd543a14`, reads log-probabilities over its declared set in 12 of its 44 annotator configurations, AlpacaEval 2.0's default (Dubois et al., 2024) among them, and at least 14 of the 805 instructions it quotes carry a closed answer format of their own. Of the 131 OpenCompass dataset configuration files that define a `GRADER_TEMPLATE` and use an LLM evaluator at revision `c4966665`, 27 quote an item with its lettered choices and parse the judge's output using its first uppercase `A` or `B` character, so a bare letter answering the quoted item is read as a verdict: `A` means correct and `B` means incorrect.[1] We ran the default MMLU judge configuration[2] with Llama-3-8B-Instruct as the judge on 500 MMLU items, each with one correct and one incorrect candidate answer: 1,000 calls in the shipped order. The grading prompt prints the gold letter. The rotation contrast uses the 500 item–candidate pairs in which reordering moves that letter between the wrong verdict's letter and C or D. The judge returns the wrong verdict on 44 calls in the former position and 1 in the latter. In the shipped order it makes 23 errors. On shipped-order calls whose gold letter is the wrong verdict's letter, errors exceed the rate on the other calls with the same candidate correctness by about 19, 82% of the 23. Every shipped-order response parses as a valid verdict. In lm-evaluation-harness, the six PISA tasks judged by a language model quote the item's lettered options with the answer they grade and map every stripped response other than `1` to `0`, so a letter is scored as incorrect,[3] while none of the 8,412 declared-set tasks (Gao et al., 2026) quotes an item with its own answer instruction before asking a different question. The checks of Section 6 belong wherever a scorer quotes what it scores.

F PUBLISHED SPECIFICATIONS, CELL BY CELL

The four specifications are measured on 64 items per cell.

The four specifications, and how far the boundary carries. AlpacaEval 2.0 is the only one of the four rendered through a chat template, so the only one whose scored context moves with the checkpoint and the only one that can discriminate: four of the five checkpoints whose contexts end at whitespace hold 0.94021 or more, `SmolLM2-1.7B-Instruct` ends at whitespace and still reads 0.42963, and the two whose contexts do not end at whitespace hold 7.01×10^{-4} or less, and on those two the declared set is absent from the five candidates the published parser reads in every one of the 64 items.

Zhuang et al.'s named checkpoint, Flan-UL2, and the Flan-T5-XL and XXL comparators have median decoder sequence masses of 0.99904, 0.99883 and 0.99864, respectively, with an option argmax on every item and identical bare and space-prefixed targets; these are sequence probabilities at a decoder reopened from `<pad>`, a different quantity from Table 11's first-token masses. Within Table 11, the counterexample is one cell of the 21 that are scorable: AlpacaEval 2.0 on `SmolLM2-1.7B-Instruct` ends at whitespace with distinct first tokens and an applicable metric, so the rule places it on the clean

[1] OpenCompass, `opencompass/datasets/generic.py`, lines 63–69 and 31–37, at the revision above.

[2] OpenCompass, `opencompass/configs/datasets/mmlu/mmlu_llmjudge_gen_f4336b.py`, at the revision above, with greedy decoding.

[3] lm-evaluation-harness, `lm_eval/tasks/pisa/utils.py`, lines 291–329.

Table 11: Declared-set first-token mass as declared / with a leading space, $10^{\text{median}(\log_{10}\mu)}$ over 64 items with linear interpolation on the log scale. G-Eval, Zhuang and Kadavath are completion prompts ending at `- Coherence:`, `Output:` and `The proposed answer is:`; AlpacaEval uses each checkpoint's chat template. Kadavath's question is the item's stem without its lettered choices or instruction, and the column scores the shared first token of `' (A)'` and `' (B)'`; their existing space makes the prefixed variant undefined. G-Eval's prefixed options share a first token on all seven checkpoints; Zhuang's and AlpacaEval's have distinct first tokens on all 64 items on every checkpoint. AlpacaEval's context ends at whitespace except on Mistral and falcon.

Checkpoint	G-Eval	Zhuang	AlpacaEval	Kadavath
Qwen3-4B-Instruct-2507	1.26e−5 / 0.99449	1.73e−6 / 0.99684	1.000 / 2.88e−14	0.48600 / —
Qwen3-8B	0.00783 / 0.96250	6.04e−4 / 0.91395	1.000 / 1.07e−10	0.82539 / —
Qwen3-14B	0.00336 / 0.99567	3.76e−5 / 0.76455	1.000 / 2.22e−15	0.67639 / —
Phi-3-mini-4k-instruct	0.0174 / 0.65558	0.11343 / 0.56764	0.94021 / 0.94021	0.82375 / —
Mistral-7B-Instruct-v0.3	2.83e−4 / 0.90844	4.83e−4 / 0.99172	9.22e−6 / 0.99426	0.39998 / —
falcon-7b-instruct	0.00606 / 0.90917	1.31e−4 / 0.96491	7.01e−4 / 0.0249	0.54744 / —
SmolLM2-1.7B-Instruct	0.0108 / 0.76328	8.59e−5 / 0.0262	0.42963 / 6.36e−5	1.39e−4 / —

side, and its median is 0.42963, beneath the floor. On these cells, the boundary condition is necessary but not sufficient. The other seven of the 28, Kadavath's column, are not scorable, because its two options share a first token on all seven checkpoints.

G-Eval's chat rendering. The G-Eval repository's system-message rendering falls below the floor on all four checkpoints with final whitespace, distinct option first tokens and an applicable first-token metric: Qwen3-4B, Qwen3-8B, Qwen3-14B and SmolLM2.

The repair's mechanism, in the same harness. `lm-evaluation-harness` provides assistant prefilling through `gen_prefix` (`lm_eval/config/task.py:39`), documented as "Prefix to start the assistant's response" at `lm_eval/api/task.py:961`. On its chat path, `build_qa_turn` creates an assistant message containing the prefix (`lm_eval/api/task.py:1104-1106`), and `add_generation_prompt=not gen_prefix` suppresses a second assistant opening (`lm_eval/api/task.py:970`). The Hugging Face backend continues that final message (`lm_eval/models/huggingface.py:1723-1728`). Our implementation instead appends the stem after rendering the generation prompt, as Appendix B shows. Both place scoring after assistant content; byte-identical rendering across implementations was not tested. Nine task files set `gen_prefix`, among them `discrim_eval_explicit` and `discrim_eval_implicit`, whose prefixes end in a quote character and so carry the boundary their empty `target_delimiter` would otherwise have dropped. We take the mechanism and supply a string; what this paper adds is where the failure sits inside its scope and how far into it the repair reaches (Table 3).

G Shared first tokens on four published prompts

Appendix F reads one G-Eval prompt. This reads all four of that instrument's dimensions on the same seven checkpoints, 100 SummEval (Fabbri et al., 2020) documents and 1600 rows per cell, 28 cells in all. The four are coherence, consistency, fluency and relevance. *Sign convention in this appendix*: $\Delta\rho = \rho(\text{bare first token}) - \rho(\text{space-prefixed sequence})$, so a positive value means G-Eval's own readout agrees with the human ratings better than the repaired one. ρ is the summary-level Spearman correlation, taken within each source document across its 16 system summaries and then averaged over the documents where it is defined; each interval is a paired bootstrap of 10 000 resamples with the document as the unit, both scoring methods recomputed on the same resample.

Constant scores and defined correlations. Space-prefixed first-token scores are constant on all 28 cells, each with 1600 rows and zero documents with a defined rank correlation for that score. G-Eval's original options have distinct first tokens on all 28 cells; their correlations range from -0.0598 to $+0.6306$.

Argmax at G-Eval's own scored position. At G-Eval's own scored position, across all four dimensions and all seven checkpoints, the argmax is one of the declared options on **0 of 44800** rows.

The 28 differences. $\Delta\rho$ moves in both directions, from -0.2194 to $+0.0831$; 19 of the 28 are negative and 9 positive, and 14 bootstrap intervals exclude zero.

What the denominators mean. The document counts are 100, 96 and 98, decided per document. The 42 missing documents are 28 on consistency and 14 on fluency across the seven checkpoints. All are missing because the human rating vector is constant there, so no rank correlation exists; none is a missing measurement and none is due to a constant model-score vector.

H WHETHER THE ORIGINAL POSITION'S ORDERING CARRIES THE FORECAST

The target is one deterministic note-only reader outcome per passage and checkpoint on the 240 RACE-H passages of Table 4. Each of Qwen3-4B, Qwen3-8B, Qwen3-14B and Meta-Llama-3-8B-Instruct writes its own note, reads it and forecasts its own reader outcome. The 960 rows contain 160, 163, 171 and 140 correct reader answers, respectively.

Writer and reader. The writer receives the passage without its question:

```
Source text:
{context}

A future reader will answer a question about this text using ONLY your
↪  note. You do not get to see the question.
Budget: at most 32 words.
Write only the note and then stop.
```

Generation is greedy with a cap of 256 new tokens. Notes end at an end-of-sequence token, contain 1 to 32 words and contain no thinking token. An over-budget or unfinished attempt is retried up to four times. Retry r, counted from 1, requests at most $\lfloor 32/(r+1) \rfloor$ words in one compact phrase. The note-only reader receives one user turn:

```
Note:
{note}

{question}
A: {option A}
B: {option B}
C: {option C}
D: {option D}
Answer with exactly one of: A, B, C, D.
```

The checkpoint's chat template opens the assistant turn, with thinking disabled and no system turn or answer stem. The reader chooses the option among `" A"`, `" B"`, `" C"` and `" D"` with the highest full-sequence log-probability (float16 forward pass, float32 log-softmax). The target y is 1 exactly when this option equals gold. The constructed-note check in Appendix C.7 compares the reader's spaced and unspaced strings on 240 passages with constructed notes, half stating the answer, on Qwen3-4B: both pick the same option on 96.25% of rows.

Forecast and scored outcome. The same checkpoint receives the note and quoted item in the forecast prompt printed in Appendix B. The phrase `Out of 100 such readers` is the question's wording; the readout is scored as a probability of the single binary reader outcome, using AUC and Brier. Readouts (a) and (b) hold the model, dtype, GPU, note, item, declared set and renormalisation fixed and move the scored position by adding the answer stem; the five options have the same first-token IDs at both positions on all four checkpoints. At the production position every passage is below the mass floor on each Qwen3 checkpoint: the median mass is 1.54×10^{-16}, 1.49×10^{-15} and 1.71×10^{-11}, and an answer letter is the argmax on 240, 233 and 239 of the 240 passages on

Qwen3-4B, 8B and 14B; with the answer stem each median exceeds 0.9999. Each forecast probability is p_S from Eq. 1, scoring the full option sequence with its tokens encoded in isolation and appended to the rendered context; on all four checkpoints, those tokens equal the in-context tokens. AUC is ROC AUC against the note-only reader's binary correctness, counting ties as one half; bootstrap intervals are the 2.5th and 97.5th percentiles of passage-bootstrap replicates (seed 0). Brier is the mean of $(p_S - y)^2$, where y is reader correctness. An average is the mean of its checkpoints' values: Table 4 averages the three Qwen3 checkpoints, and Table 13 gives that average and the average of all four. An AUC difference is the readout minus its reference; a Brier difference is the reference minus the readout, so positive values favour the readout.

Table 4 in Section 6 reports these readings.

Table 12: Constant-forecast Brier baselines on the same evaluation rows as Table 4. The fitted constants use each checkpoint's empirical reader-correctness rate; the Qwen3 average is the mean of its three checkpoints. These are descriptive baselines fitted on the evaluation rows.

Forecast	Qwen3-4B	Qwen3-8B	Qwen3-14B	Qwen3 average	Llama-3-8B-Instruct
Constant 0.5	0.2500	0.2500	0.2500	0.2500	0.2500
Fitted constant	0.2222	0.2179	0.2048	0.2150	0.2431

The stemmed position's Qwen3-averaged and Llama Brier scores each exceed both descriptive constant baselines.

The original position's Qwen3-averaged AUC interval contains 0.5, so its item-level ranking is not distinguishable from chance.

In Table 4's Qwen3 rows, the two positions select the same grid value on 4.6%, 9.2% and 12.9% of rows, against 96.67%, 96.25% and 98.33% for the reader's spaced and unspaced options on the same rows (Appendix C.7); the original position puts `10` on 223, 236 and 187 of its 240 rows, while the repaired one spreads across the grid.

Five readouts on the four checkpoints. Table 13 compares (a) the production prompt without a stem; (b) that prompt followed by `Out of 100 such readers, about ` ; (c) that prompt with the quoted item's closing instruction deleted, without a stem; (d) Appendix I's T1 with the original ending, without a stem; and (e) T1 followed by `Estimate: ` . The note and reader target are fixed across readouts, and all intervals use the same 20,000 passage-cluster bootstrap replicates across readouts and checkpoints. The Qwen3 checkpoints are read here a second time, on other hardware than Table 4's Qwen3 rows. The second reading differs from Table 4 in Qwen3-averaged AUC by -0.0002 for (a) and $+0.0004$ for (b) at the printed precision, by at most 0.0011 on a single checkpoint, and in Brier by at most 0.0002. Llama-3-8B-Instruct was read at snapshot `8afb486c`, whose weights, tokenizer and configuration files are the same blobs as those of Appendix I's `e1945c40`: of the 14 files both cached snapshots hold, only the repository's model card differs. Deleting the instruction returns the median option mass above 0.9998 on each Qwen3 checkpoint but not on Llama-3-8B-Instruct, whose median is 0.0023. Against the stem, the averaged Brier score is worse for (c), (d) and (e), each interval below zero.

The number the model writes. Readout (f) generates greedily from (a)'s prompt, at most 48 new tokens, and reads the first maximal digit run that is one of the five options, skipping any other number; the forecast is that value divided by 100, and a generation with no grid number is scored 0.5. A grid number is found on 239, 239, 240 and 116 of the 240 passages on Qwen3-4B, Qwen3-8B, Qwen3-14B and Llama-3-8B-Instruct, whose other 124 generations answer the question in prose. The forecasts concentrate on one value: 50 on 167 and 221 of Qwen3-4B's and Qwen3-8B's passages, 70 on 232 of Qwen3-14B's. Their AUCs are 0.5775, 0.5146, 0.4974 and 0.5529, averaged over the four 0.5356 $[0.5083, 0.5628]$; against (a) the averaged difference is $+0.0353$ $[-0.0155, +0.0857]$ and against (b) -0.0772 $[-0.1169, -0.0355]$. On the Qwen3 rows with a grid number, against (b), it is -0.0847 $[-0.1304, -0.0378]$. The averaged Brier score of (f), 0.2448, is lower than (b)'s by 0.0089 $[-0.0040, +0.0222]$ and higher than the fitted constant's 0.2220: a forecast held near one value scores near a constant on Brier without ranking the passages.

Table 13: Five forecast readouts on the same 240 passages per checkpoint. The upper panel gives AUC with its 95% interval and Brier for the Qwen3 and four-checkpoint averages, each with equal checkpoint weights. Paired differences in the lower panel use the four-checkpoint average and $\Delta\mathrm{AUC} = \mathrm{AUC}(x) - \mathrm{AUC}(r)$ and $\Delta\mathrm{Brier} = \mathrm{Brier}(r) - \mathrm{Brier}(x)$, so positive values favour x. All intervals use a paired passage-cluster bootstrap shared by every readout and checkpoint. Readouts (a)–(e) are defined in the text above the table.

	Qwen3 average		Four-checkpoint average	
Readout	AUC [95% interval]	Brier	AUC [95% interval]	Brier
(a)	0.4927 [0.4388, 0.5457]	0.4748	0.5003 [0.4544, 0.5452]	0.4221
(b)	0.6156 [0.5607, 0.6689]	0.2543	0.6128 [0.5634, 0.6605]	0.2538
(c)	0.5651 [0.5083, 0.6218]	0.2713	0.5640 [0.5148, 0.6124]	0.2644
(d)	0.5926 [0.5381, 0.6460]	0.2930	0.5879 [0.5404, 0.6342]	0.2800
(e)	0.6015 [0.5445, 0.6570]	0.2933	0.6065 [0.5564, 0.6550]	0.2823

Reference r	Readout x	ΔAUC [95% interval]	ΔBrier [95% interval]
(a)	(b)	$+0.1125\,[0.0477, 0.1766]$	$+0.1683\,[0.1412, 0.1951]$
(a)	(c)	$+0.0637\,[0.0056, 0.1218]$	$+0.1577\,[0.1347, 0.1800]$
(a)	(d)	$+0.0876\,[0.0243, 0.1501]$	$+0.1421\,[0.1204, 0.1628]$
(a)	(e)	$+0.1062\,[0.0422, 0.1709]$	$+0.1398\,[0.1181, 0.1610]$
(b)	(a)	$-0.1125\,[-0.1766, -0.0477]$	$-0.1683\,[-0.1951, -0.1412]$
(b)	(c)	$-0.0489\,[-0.0791, -0.0191]$	$-0.0106\,[-0.0202, -0.0007]$
(b)	(d)	$-0.0250\,[-0.0518, 0.0019]$	$-0.0263\,[-0.0418, -0.0104]$
(b)	(e)	$-0.0063\,[-0.0291, 0.0160]$	$-0.0286\,[-0.0422, -0.0145]$

Short answer stems. On the three Qwen3 checkpoints, the forecast is also read after `Number:` and after `Forecast:` , in the same run as Table 13's readouts (a) and (b), whose values it reproduces. Each stem is three tokens at the end of the rendered context and leaves the five options' first tokens unchanged. After either stem the option mass has a median of at least 0.997 on each checkpoint, with at most 6 of 240 passages below the floor. Readout (b) reaches AUC 0.6286, 0.6446 and 0.5738 on Qwen3-4B, 8B and 14B; `Number:` reaches 0.6052, 0.6620 and 0.5763, and `Forecast:` 0.6082, 0.6461 and 0.5674. Averaged over the three, `Number:` is $+0.1218$ $[+0.0500, +0.1935]$ above readout (a) and -0.0011 $[-0.0213, +0.0191]$ from readout (b); `Forecast:` is $+0.1145$ $[+0.0414, +0.1872]$ above (a) and -0.0084 $[-0.0328, +0.0162]$ from (b).

I THE READING UNDER EIGHT PROMPT CONDITIONS

The probe and what is held fixed. The probe repeats Table 3's measurement on its seven checkpoints at the snapshots in Appendix A.2, and on `meta-llama/Meta-Llama-3-8B-Instruct` at `e1945c40`, loaded from a cache on the compute host. Each condition uses 64 items per checkpoint: the same first 64 RACE-H items in frozen order, except for D1's item replacement. The five forecast options remain `10`, `30`, `50`, `70`, `90`. Every reading uses a float16 forward pass, float32 log-softmax and the checkpoint's own chat template through its generation prompt, with `enable_thinking=False` where the template reads that argument. All eight conditions were read on one GPU model.

C0 reads the production prompt again, with Table 3's fixed note, `Landlord only gives when told to take, not give.` The user-turn forecast text and the quoted item's closing letter instruction are those printed in Appendix B; C0 changes that appendix's example note to this fixed note. N1 changes the note, S1 the turn containing the forecast instruction, and D1 the dataset and item format. A1 and A2 insert sentences into C0, and T1 and T2 replace its whole user-turn template. The fixed note is used in every condition except N1.

The mass without an answer stem is read at the original ending. The answer stem is `Out of 100 such readers, about` for C0, N1, S1, D1, A1 and A2. T1 and T2 are read with their own answer stems, printed below, and also with the original answer stem. On Mistral and falcon, whose templates omit final whitespace, the separator control appends one space to the rendered prompt. Each prompt also receives a greedy continuation of at most 16 new tokens from the original ending.

Each condition uses the collapse definition in Section 3 at the original ending; letter pull means that at least half the items have one of their own answer letters as the unrestricted argmax; repair means a median at or above the floor with the condition's primary answer stem. The letter count strips whitespace and a tokeniser's leading word-boundary marker. Letter pull excludes Llama-3-8B-Instruct's `A` when its continuation begins `A clever`, `A classic` or `A nice`: these are prose openings. Table 15 separates the mechanisms.

N1: per-item notes. N1's notes are the ones `Qwen3-4B-Instruct-2507` wrote for these 64 items in a separate reading with a twelve-word budget; the same 64 notes replace C0's fixed note for every checkpoint. All other prompt text is C0's.

S1: system and user turns. The system text is exactly:

```
You will be given a note and a quoted question. A future reader will be
↪  shown ONLY the note, not the source text, and will then be asked the
↪  quoted question. Out of 100 such readers, how many answer correctly?
↪  Reply with exactly one of: 10, 30, 50, 70, 90.
```

The user text is exactly the following template, with {`note`} replaced by the fixed note and {`q`} by the unchanged RACE-H item block, including its closing answer instruction:

```
Note:
{note}

Quoted question:
{q}
```

Mistral's chat template merges the system text into the user turn on all 64 items. The other seven templates retain it in a separate block on all 64 items each; all eight therefore have S1 readings.

D1: MMLU-Pro items. D1 uses the first 64 rows of the MMLU-Pro test split in stored order (Wang et al., 2024c). There are 62 items with ten options, one with nine and one with seven. The item block replacing {`q`} has exactly this form:

```
{question}
{rows}
Answer with exactly one of: {letters}.
```

Here {`question`} is the row's question, {`rows`} is its options joined by single newlines, each as `<letter>: <option text>`, and {`letters`} is that row's letters joined by comma-space. The letters start at A and end at the item's last option. The forecast template and fixed note remain C0's.

A1 and A2: added sentences. A1 inserts `The question's instruction to answer with a letter is for the readers.` and one following space immediately before `Out of 100 such readers, how many answer correctly?` in C0. A2 adds `Your task is to forecast their number of correct answers.` and one following space immediately after the first insertion. Both sentences come from T1 in that order; the answer stem and all other text remain C0's.

T1 and T2: whole forecast templates. A language model, `gpt-6-astra`, wrote both templates from a description of the forecasting task and a made-up example item, with no information about the failure or the checkpoints; that task text is printed below in full. It was asked for two complete user-turn templates with {`note`} and {`q`} placeholders and an answer stem for each. The only checks were mechanical: the JSON parsed, each placeholder occurred once, there were no other placeholders, and neither stem contained a digit. Both templates were used as written. In each, {`note`} is the fixed note and {`q`} is the unchanged RACE-H item block. Each assigns the letter instruction to the readers and ends with an output contract. As in Appendix B, arrows in these boxes mark typeset line breaks rather than newlines in the prompt.

The task given to the writer, in full:

```
# Write two prompt templates

We need prompts for the following forecasting task, to be sent as the user message to
several open instruction-tuned chat models through each model's own chat template.

The situation: someone read a text passage and wrote a short note about it (a sentence or
two). Later, a group of readers will be shown ONLY that note, never the passage, and will
then answer one multiple-choice question about the passage. The question is shown to them
exactly as it appears in the block below, including the answer instruction at its end.

The model we prompt sees the note and that same question block. It must forecast how many
out of 100 such readers answer the question correctly, and reply with exactly one of the
five values 10, 30, 50, 70, 90.

A question block looks like this (a made-up example; real ones come from a
reading-comprehension dataset and always have this shape):

    What does the writer mainly want to tell us?
    A: Travel is expensive for students.
    B: Small towns are worth visiting.
    C: Trains are faster than buses.
    D: Maps are often out of date.
    Answer with exactly one of: A, B, C, D.

Write TWO templates for this task, written the way you would write them for real use.
Make them differ from each other in structure and wording, not only in small phrasing.
Requirements:
- Use the placeholders `{note}` for the note text and `{q}` for the question block. `{q}`
  must appear exactly once and be inserted unchanged; do not rewrite or reformat it.
- Use no other placeholders. Plain text only.
- The template is the whole user message. Do not add a system message.
- The reply the model gives must be exactly one of 10, 30, 50, 70, 90.
- For each template, also give an answer lead-in: the first words of an ideal assistant
  reply, ending right before the number (for example `Estimate: `). It must end with a
  space or a colon-space and must not contain a number.

Write the result to `templates.json` in this directory, as
`[{"id": "T1", "template": "...", "answer_lead_in": "...", "notes": "..."}, {"id": "T2",
↪  ...}]`,
and check that the file parses as JSON and that each template contains `{note}` and `{q}`
exactly once and no other `{...}`.
```

T1, in full:

```
Someone read a passage and wrote the short note below. Imagine 100
↪  readers who will see only this note and the question below,
↪  including its final answer instruction. They will never see the
↪  passage. Each reader will answer the question, and correctness will
↪  be judged against the original passage.

Note:
{note}

Question:
{q}

Estimate how many of these 100 readers will answer correctly. Consider
↪  how much useful information the note provides, whether it
↪  distinguishes the choices, and where readers may misunderstand or
↪  have to guess. The question's instruction to answer with a letter is
↪  for the readers. Your task is to forecast their number of correct
↪  answers.

Choose the closest of these five counts to your estimate: 10, 30, 50,
↪  70, 90. Reply with exactly one of these values and nothing else.
```

T1's answer stem is `Estimate: `.

T2, in full:

```
OUTPUT CONTRACT
Return one bare value from this set: 10, 30, 50, 70, 90.

FORECAST TARGET
Number of correct answers among 100 readers taking a
↪  reading-comprehension test. Each reader receives only a short note
↪  written by someone who read the passage, plus the complete test item
↪  below. Neither you nor the readers receive the passage. The
↪  passage's answer key determines correctness.

TEST ITEM SHOWN TO READERS
BEGIN TEST ITEM
{q}
END TEST ITEM

NOTE AVAILABLE TO READERS
BEGIN NOTE
{note}
END NOTE

ASSESSMENT PROCEDURE
1. Identify what information the item asks readers to recover from the
↪  passage.
2. Assess which choices the note supports or rules out. Account for
↪  missing or ambiguous information, plausible mistakes, and guessing.
3. Estimate how many readers will select the correct answer, then select
↪  the nearest permitted count.

Keep the assessment to yourself. The letter-answer instruction inside
↪  the test item governs the readers' answers. Your response must
↪  follow the output contract: exactly one of 10, 30, 50, 70, 90, with
↪  no label, explanation, punctuation, or percent sign.
```

T2's answer stem is `Predicted number correct: .`

The readings. Tables 14 and 15 are computed from the per-item in-context masses and argmax tokens; the letter-pull count also reads the opening of each Llama continuation. Medians are displayed to five decimal places or four significant figures in scientific notation; counts use the unrounded values. T1 and T2 remove letter pull on every checkpoint; the three below-floor cells in each are Mistral's and falcon's omitted separators, plus Phi-3's shared first token under T1 and SmolLM2 under T2. With the stem, Llama-3-8B-Instruct's C0 median is 0.99352.

Table 14: Eight prompt conditions on 64 items per checkpoint. Each cell gives the raw in-context median at the original ending, followed by the letter-argmax count and below-floor count, separated by a semicolon; both counts are out of 64. † marks shared first tokens. Llama's counted letters are the article `A`. Under S1, Mistral's template merges the system text into the user turn. Checkpoint names are abbreviated from Table 3, with Llama-3-8B-Instruct added. With each condition's own answer stem, every median is at or above the floor except falcon's under D1, 0.45421, and T2, 0.33207.

Checkpoint	C0	N1	S1	D1	A1	A2	T1	T2
Qwen3-4B	2.707e−15 63;64	2.669e−16 64;64	0.00404 48;53	2.761e−10 61;64	5.824e−7 56;61	0.99991 5;9	1.00000 0;0	1.00000 0;0
Qwen3-8B	8.144e−14 61;64	1.183e−14 61;64	0.93951 25;25	3.107e−11 14;64	0.00900 42;50	0.99895 1;2	1.00000 0;0	0.99963 0;0
Qwen3-14B	1.371e−8 62;64	4.235e−11 63;64	1.404e−4 60;61	2.718e−11 39;64	5.807e−12 62;64	0.00218 54;60	1.00000 0;0	1.00000 0;0
Phi-3-mini	5.801e−4† 63;64	9.051e−4† 61;64	0.00137† 64;64	0.00271† 18;64	0.00169† 63;64	0.03618† 61;64	0.39024† 20;46	0.53902† 2;26

Table 14 continued

Checkpoint	C0	N1	S1	D1	A1	A2	T1	T2
Mistral v0.3	3.488e−6 30;64	1.907e−6 31;64	5.023e−7 57;64	5.412e−7 17;64	6.074e−6 3;64	4.694e−6 0;64	2.005e−6 0;64	1.180e−7 0;64
falcon	1.051e−4 0;64	1.274e−4 0;64	3.158e−4 0;64	6.909e−5 0;64	1.108e−4 0;64	9.853e−5 0;64	1.385e−4 0;64	6.051e−5 0;64
SmolLM2	0.39017 46;53	0.41918 41;52	0.01977 64;64	0.08264 57;62	0.48962 23;34	0.65508 1;1	0.82232 0;0	0.31081 7;64
Llama-3-8B	2.565e−4 49;64	1.113e−4 8;64	0.58728 0;21	2.626e−4 47;64	1.141e−5 46;64	1.128e−5 49;64	0.95180 0;0	0.84624 0;0

Table 15: What holds the scored position at the original ending in each condition of Table 14, counted over its eight checkpoints. *Answer letters*: one of the item's own letters is the unrestricted argmax on at least 32 of 64 items, not counting Llama-3-8B-Instruct's article `A`. *Prose opening*: a word token that opens prose, including `Answer` and that article, is the argmax on at least 32. *Omitted separator*: the rendered ending lacks the whitespace the options need. *Shared first token*: the options' in-context first tokens coincide. *Options*: median mass at or above the floor with distinct first tokens. Rows below the first can overlap. The last row counts repairs.

	C0	N1	S1	D1	A1	A2	T1	T2
Median below the floor	8	8	6	8	8	5	3	3
Answer letters	5	5	5	3	4	2	0	0
separator present, first tokens distinct	4	4	3	3	3	1	0	0
Omitted separator	2	2	2	2	2	2	2	2
Shared first token	1	1	1	1	1	1	1	1
Prose opening	1	1	0	4	1	1	0	0
Below the floor, none of the above	0	0	0	0	1	0	0	1
Options hold the position	0	0	2	0	0	3	5	4
Repaired by the condition's own stem	8	8	8	7	8	8	8	7

C0 compared with Table 3. C0 retains all three classifications on the original seven checkpoints. Qwen3-8B, Qwen3-14B, Mistral and falcon were first read on C0's GPU model and agree item by item. Qwen3-4B, Phi-3-mini and SmolLM2 were first read on another GPU model; their medians change by at most 0.19% and their minima by at most 3.3%, and SmolLM2's below-floor count changes from 50/64 in Table 3 to 53/64. All letter rates and stemmed below-floor counts agree.

A larger checkpoint. `Qwen/Qwen3-32B` at snapshot `9216db57` was read under C0 and in the instruction experiment of Section 4, on another GPU model than the seven checkpoints of Table 3. At the production position its median in-context mass is 3.79×10^{-3} (minimum 5.91×10^{-7}), all 64 items are below the floor, and no item has a forecast option at the argmax: the argmax is an answer letter on 35 items (rate 0.547) and `Answer` on 29. With the stem the median is 0.99928 (minimum 0.87448) and no item is below the floor. Deleting the imperative takes the letter rate to 0.000 and the median to 0.99085, with a digit at the argmax on all 64. Of its C0 continuations, 48 answer the item first, 13 of them after `Answer:`, and the other 16 open with `Answer:` followed by a number (Table 16). The token-class prediction holds for the prose and numeric substitutes and fails for the list, translation and count substitutes, where a digit takes the position on 38, 49 and 64 of 64 items.

What the unrestricted continuation writes. Table 16 classifies the C0 continuations: the production prompt with the fixed note and the original ending, no answer stem, greedy decoding of at most 16 new tokens. Fixed rules on the opening, regular expressions and string comparisons with the item's question and options, assign the first class that matches: an answer to the item, a forecast, a recital of the option list, the question or an instruction, other prose, or other. A continuation that opens as prose and then answers the item, such as "`Based on the note, the correct answer is D:`", "`D is not true, as the note`" or "`The text is mainly about B:`", counts as prose; a separate flag, not tabulated, for an answer letter anywhere in the text catches the first and the third. Text after a special token or a new chat turn is dropped. When a continuation stops at the token limit after a single line labelled with an item letter, the rules cannot tell an answer from the

first line of an option list, and they count it as an answer. A forecast that comes after a long answer can also be cut off, so both flag columns are lower bounds, most visibly on Phi-3-mini (27 of 63). Three people labelled 100 of the 512 continuations of the eight checkpoints other than Qwen3-32B by what each does first, with the checkpoint and the rule class hidden. The sample is stratified by rule class and checkpoint and contains the three continuations quoted above; on ten practice records with fixed answers each annotator scored 10 of 10. They agree with each other on 99 of the 100 (Fleiss' $\kappa = 0.991$), and their majority label matches the rule class on 92, or on 97.9% when the sample is reweighted to all 512. Seven of the eight differences are continuations the rules class as prose and all three annotators label an answer to the item: the three quoted above and four falcon continuations that state an option's content as the answer. The eighth is a falcon line labelled with an item letter at the token limit, which the rules count as an answer and the annotators as the start of a recited list. None of the eight is on a Qwen3 checkpoint. By the annotators' reading, the rules undercount answers to the item.

Table 16: Classes of the C0 continuations, greedy, at most 16 new tokens, on the same 64 items for each checkpoint. The first five columns partition the items by the opening; the last two are flags that overlap them. "Answer, then grid": the opening answers the item and a grid value follows it. "Grid anywhere": a grid value anywhere in the text, outside a recited option list. A continuation that gives a forecast first can give a value off the grid.

Checkpoint	Answers first	Forecast first	List or question	Prose	Other	Answer, then grid	Grid anywhere
Qwen3-4B-Instruct-2507	63	1	0	0	0	63	63
Qwen3-8B	63	0	0	1	0	61	61
Qwen3-14B	62	0	0	2	0	62	62
Phi-3-mini-4k-instruct	63	0	0	1	0	27	27
Mistral-7B-Instruct-v0.3	29	30	0	5	0	2	28
falcon-7b-instruct	3	35	18	8	0	0	35
SmolLM2-1.7B-Instruct	46	18	0	0	0	0	18
Meta-Llama-3-8B-Instruct	4	1	0	59	0	0	1
Qwen3-32B	48	16	0	0	0	38	54

The continuations and the counted tokens. Every letter counted for Meta-Llama-3-8B-Instruct is the article in `A clever question!`, `A classic` or `A nice`. These are prose openings, rather than answers to the item.

Phi-3-mini's raw mass sums a shared first token (id `29871`) in every condition and item. Those masses are not separability readings, including its T1 collapse. Its stemmed first tokens are distinct on every item in all eight conditions. SmolLM2 under T2 holds a raw median of 0.31081, below the floor on all 64 items, with an option at the argmax on 56/64.

The omitted separator. Mistral and falcon collapse under all eight conditions at the original ending. With the separator supplied, Mistral's median exceeds the floor in all eight. Falcon's medians with the separator supplied are $0.384\,82$, $0.394\,38$, $0.697\,28$, $0.339\,24$, $0.344\,99$, $0.375\,06$, $0.448\,11$, $0.149\,91$ in the condition order used above, and its option-argmax counts are 63, 63, 64, 61, 63, 63, 62, 40, all out of 64. Thus an option is at falcon's argmax on most items in every condition, while its median remains below the floor in seven; S1 clears it.

The original answer stem on T1 and T2. For comparison with their own stems, T1 and T2 are also read with `Out of 100 such readers, about .` This gives eight repairs on T1 and seven on T2, with falcon below the floor on T2, as with the templates' own stems. No item has an item-letter or A–D answer-letter argmax with the primary stem in any of the eight conditions on the eight checkpoints, with the original answer stem on T1 and T2, or in the additional Qwen3-32B C0 reading.