

Lost in Choice: Evaluating LLMs Under Extreme Multiple-Choice Scaling

Alessio Cocchieri* Luca Ragazzi* Giuseppe Tagliavini Gianluca Moro*

{a.cocchieri, l.ragazzi, giuseppe.tagliavini, gianluca.moro}@unibo.it

Department of Computer Science and Engineering, University of Bologna, Italy

Abstract

Existing multiple-choice evaluations use small option sets and thus conflate several sources of failure when candidate spaces grow. We provide a controlled factorial decomposition of large-choice degradation across four axes: whether a question requires comparing its options, candidate cardinality (up to 1,000), spatial layout, and gold-answer position. Across six reasoning LLMs, comparative questions degrade far more than questions solvable without inspecting the options; dispersion amplifies this gap; and accuracy often collapses at late-middle and terminal positions rather than recovering in the classic lost-in-the-middle U-shape. We call this asymmetric, structure-dependent failure mode *Lost in Choice*. A replication on 500 real ToolBench tool-selection trajectories per condition reproduces the same hardest position on two additional model-task evaluations. Finally, an audit of reasoning traces yields a taxonomy of the heuristics and shortcuts models adopt as candidate spaces expand.¹

1 Introduction

As Large Language Models (LLMs) are increasingly deployed as decision-making systems, their ability to reliably select among large sets of competing candidates has become a central requirement. This need emerges naturally in modern applications: tool retrieval libraries now comprise over 16,000 APIs (Shi et al., 2025), while biomedical annotation, structured coding, and knowledge-graph completion operate over large taxonomies or structured candidate spaces (di Lena et al., 2015; Ferrari et al., 2022; Molfetta et al., 2025)—making high candidate cardinality a structural constraint.

Concurrently, recent architectural advances have expanded input context windows to millions of

*Equal contribution (co-first authors).

¹Code and data are publicly available under the MIT license and CC BY 4.0, respectively: <https://disi-unibo-nlp.github.io/lost-in-choice/>

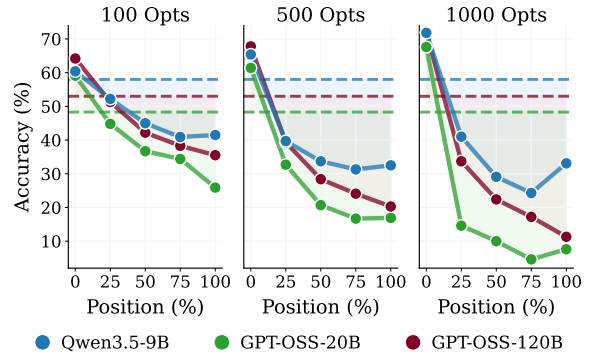


Figure 1: **The Lost-in-Choice phenomenon.** Accuracy degradation as the number of options scales (100, 500, and 1000) across five gold-answer positions. Dashed lines indicate each model’s baseline on 10 options.

tokens (Peng et al., 2024), technically permitting models to process thousands of alternatives in a single forward pass. Yet, the literature on long-document processing reveals that LLMs exhibit a “lost-in-the-middle” phenomenon (Liu et al., 2024), where information retrieval degrades when critical data resides far from the input boundaries. Whether a similar bottleneck plagues multiple-choice decision-making—and whether it stems from a simple failure to retrieve a known answer or a deeper inability to compare candidates as the decision space expands—remains an open question.

In this paper, we bridge this gap through a controlled *factorial* analysis—a design that systematically crosses candidate cardinality, layout, and position. To isolate the nature of model failures, we introduce a taxonomy of multiple-choice tasks. We distinguish between **Option-Independent** samples, where the answer can be derived from internal knowledge prior to inspecting candidates (e.g., “What is the capital of Italy?”), and **Option-Dependent** samples, where the correct choice emerges only through a comparative analysis of the available alternatives (e.g., “Which of the following algorithms is most efficient?”).

To evaluate these dynamics, we construct **MMLU-XT** (eXTreme), a curated evaluation benchmark of 1,000 samples across four core domains derived from MMLU-Pro (Wang et al., 2024). With this framework, we scale candidate options from the baseline 10 up to 1,000 by injecting controlled, in-domain distractors. Crucially, we manipulate the spatial arrangement of the original 10-option set, organizing them either as a contiguous Block or as a Sparse configuration interleaved across the context. Further, we slide the position of the gold answer across five window locations (from 0% to 100% of the candidate list) to map the combinatorial interaction between option layout, serial-position effects, and decision complexity.

Our contributions are summarized as follows:

- We separate Option-Independent retrieval from Option-Dependent comparison and uncover a *Reasoning vs. Context Paradox*: for four out of six models, 1,000 localized options on the former are less damaging than the native 10-option version of the latter.
- We identify *Lost in Choice* through a four-axis decomposition: degradation depends jointly on comparison load, cardinality, layout, and position. Unlike the classic U-shape, performance generally fails to recover near the end of the list (Figure 1).
- We present MMLU-XT, a controlled 31,000-instance evaluation suite, and replicate its positional collapse on real ToolBench tool-selection trajectories for two models.
- We establish a behavioral taxonomy of LLM decision strategies and audit how model reasoning shifts as the option space expands.

2 Related Work

2.1 Multiple-Choice Evaluation

Multiple-choice benchmarking remains a central paradigm for evaluating LLMs (OpenAI, 2023), providing a controlled setting in which models must discriminate among competing candidates (Balepur et al., 2025). As performance approaches saturation on traditional 4-option benchmarks, recent efforts have increased the number of candidates to better differentiate model capabilities. MMLU-Pro (Wang et al., 2024) extends the original MMLU (Hendrycks et al., 2021) to 10 options, a design also adopted by MedXpertQA (Zuo et al.,

2025) to stress-test expert-level reasoning. More recent studies push this scaling further, expanding candidate sets to 32 (Fan et al., 2025) and even 100 options (Lee and Son, 2026), while revealing increasingly pronounced positional artifacts. The possibility that larger candidate sets may impair rather than improve decision quality also has a precedent in behavioral science: research on choice overload shows that expanding an assortment can sometimes reduce engagement and decision quality (Iyengar and Lepper, 2000). We do not posit a shared cognitive mechanism between humans and LLMs; rather, this literature motivates the analogous empirical question of whether increasing option cardinality can itself degrade model performance.

Prior work has also documented that LLM performance is highly sensitive to seemingly minor changes in the candidate set. For instance, accuracy can drop substantially when moving from 4 to 5 options in MedQA (Jin et al., 2020; Frisoni et al., 2024). Model predictions are also strongly influenced by the alternatives presented, with LLMs often selecting the “least incorrect” option rather than uniquely identifying the correct answer (Wang et al., 2025). Forced-choice evaluation can further mask models’ inability to abstain when no valid answer is available, even when abstention is explicitly presented as the only option that avoids putting a patient’s life at risk (Cocchieri et al., 2026a). Such findings suggest that models struggle to maintain stable discriminative boundaries across option variations (Zhao et al., 2025). This fragility is further reflected in ReMedQA (Cocchieri et al., 2026b), where high baseline accuracy can mask sensitivity to minor option perturbations. More broadly, studies of LLM sycophancy (Ranaldi and Pucci, 2023) show that decision reliability can deteriorate under misleading authoritative context and superficial formatting perturbations (Molfetta et al., 2026).

Yet these studies do not isolate option cardinality from semantic difficulty, spatial layout, and answer position. We address this gap through a controlled factorial design that separates these dimensions and complements outcome-level evaluation with an analysis of the resulting reasoning trajectories.

2.2 Positional Bias in LLMs

A complementary line of research investigates positional biases in long-context processing. In cognitive psychology, the serial-position effect describes the tendency to recall items at the beginning and end of a list more accurately than those

in the middle (Murdock Jr, 1962). LLMs exhibit a superficially similar performance pattern, commonly known as the “lost-in-the-middle” phenomenon (Liu et al., 2024): relevant information is used more reliably when it appears near the beginning or end of the context, whereas performance degrades at intermediate positions. Increasing the physical distance between relevant pieces of information to retrieve can further exacerbate these failures, indicating difficulty in integrating evidence distributed across an extended context (Tian et al., 2025). Position-dependent effects have been observed across tasks, including in-context learning (Lu et al., 2022), ranking (Tang et al., 2024), and text summarization (Ravaut et al., 2024). These long-input limitations have motivated alternative processing strategies over long documents, including chunking texts into semantically coherent segments (Moro and Ragazzi, 2022, 2023), and extracting or pruning salient information before downstream processing (Moro et al., 2023; Ragazzi et al., 2024, 2025; Italiani et al., 2025).

Within multiple-choice evaluation, positional bias has primarily been studied as a systematic preference for particular answer indices—for example, selecting option “A” disproportionately—that can persist independently of option content (Pezeshkpour and Hruschka, 2024; Zheng et al., 2024; Li and Gao, 2025; Yang et al., 2025). This index-level perspective leaves open a different question: *how does the spatial organization of the entire candidate set affect the comparisons required to identify the correct answer?* We address this question by manipulating both the layout of the options and the position of the correct candidate, thereby connecting positional sensitivity in long-context processing with comparative decision-making in multiple-choice evaluation.

3 Benchmark Construction

We introduce **MMLU-XT**, a diagnostic benchmark derived from MMLU-Pro (Wang et al., 2024) designed to test LLMs under large candidate spaces. To this end, we manipulate the core questions along three primary dimensions: (i) cardinality, (ii) spatial disposition, and (iii) target position.

3.1 Question Typology

To decouple core knowledge retrieval from option comparison complexity, we formalize a key distinction in question typology:

- **Option-Independent:** Questions solvable via internal knowledge prior to choice inspection (e.g., “What is the capital of Italy?”).
- **Option-Dependent:** Questions requiring a comparative analysis of the option set (e.g., “Which of the following is most efficient?”).

Classification is done by majority vote among three LLM judges and validated against human annotations; full protocol details and examples across domains are provided in Appendix A.1. To span diverse reasoning modalities, we select instances from four main MMLU domains—Computer Science, Math, Law, and Health—sampling 250 questions each, for a total of 1,000 unique questions.

3.2 Option Scaling

To assess how search-space expansion impacts reasoning stability, we embed the original 10-option sets within in-domain distractor pools, yielding expanded option sets of size $N \in \{100, 500, 1000\}$.

Random in-domain distractors serve two purposes. First, because they are interchangeable, rearranging them changes layout and position without systematically changing semantic difficulty, allowing us to isolate how models respond to structural variation. Second, they extend the original plausible candidates with a weaker tail, reflecting how genuinely competitive alternatives become scarcer as candidate pools grow. Hard-negative pools would instead test how structural effects interact with dense semantic competition; Sections 5.5 and 7 discuss this complementary setting. To maintain cross-scale consistency, these pools adhere to a strict prefix property: $N_{100} \subset N_{500} \subset N_{1000}$.

To ensure the resulting pool is free from label noise or coincidental correctness, we screen all 990,000 unique question-distractor pairs (990 distractors $\times$ 1,000 source questions) via a dual-judge pipeline (GPT-OSS-120B and Gemini-3-Flash). We resolve flagged options (affecting 51.5% of questions) through a strict replacement and fallback protocol. Two authors independently audited 200 pairs (100 flagged and 100 passed), with Cohen’s $\kappa = 0.82$. The screening attained 94.0% precision in its weakest domain (Law) and a global false-negative rate below 4.5%; the complete correction and fallback protocol is provided in Appendix A.2.

3.3 Spatial Disposition

To analyze how the spatial distribution of candidates influences reasoning, we design two place-

Domain	Opt-Indep.	Opt-Dep.	Total
Computer Science	204	46	250
Health	107	143	250
Law	57	193	250
Mathematics	172	78	250
Questions	540	460	1,000
Variations (3 Card. $\times$ 2 Lay. $\times$ 5 Pos.)			$\times$ 30
Baseline Configuration (10-Opt)			+1
Total Instances			31,000

Table 1: **MMLU-XT evaluation scale.** Breakdown of core questions across domains and their systematic expansion to 31,000 unique test instances.

ment paradigms for the original 10-option set:

- **Block Mode:** The 10 original options remain contiguous within the candidate space, preserving localized signal density.
- **Sparse Mode:** The 10 original options are distributed uniformly across 10 segments of size $N/10$ and interleaved by distractors, maximizing spatial dispersion.

3.4 Target Position

To map positional biases, we sweep the relative position of the target options across the context window using $w \in \{0, 0.25, 0.5, 0.75, 1\}$. For a pool of size N and an original answer index $k \in \{0, \dots, 9\}$, the gold answer is assigned the absolute index $I_{\text{gold}} = \lfloor w \cdot (N - 10) \rfloor + k$.

This formulation ensures equidistant coverage of the context window while introducing a local jitter proportional to k , preventing the exploitation of fixed-index biases by keeping the target within a controlled ± 10 neighborhood of w .

3.5 Combinatorial Scale

The cross-product of our evaluation design variables (3 Cardinalities $\times$ 2 Layouts $\times$ 5 Positions) yields 30 distinct variants per question. Including the original $N = 10$ baseline, the final evaluation suite has exactly 31,000 unique test instances (see Table 1 for detailed statistics).

4 Evaluation Framework

4.1 Models

To manage the computational footprint of evaluating 31,000 unique instances per model while ensuring broad architectural representation, we curate a suite of six state-of-the-art LLMs:

- **Open-Source:** We evaluate **GPT-OSS (20B, 120B)** (OpenAI, 2025), **Qwen3.5 (9B, 27B)** (Qwen Team, 2026), and **Gemma-4-31B** (Gemma Team, 2026).
- **Closed-Source:** We use **Gemini-3-Flash** as a black-box baseline specifically optimised for long-context efficiency and robust reasoning.

All six LLMs are *reasoning* models (Guo et al., 2025): they emit an extended CoT trace enclosed in thinking tags (e.g., `<think>...</think>`) before producing a final answer (Wei et al., 2022). This slow-thinking paradigm, in which the model can explore multiple solution paths before committing, is well suited to our challenging setting: it maximizes the model’s opportunity to exercise genuine discriminative reasoning rather than superficial pattern matching across 100–1000 candidates. To eliminate parsing noise, we enforce structured output for the final response and extract the answer index. Further details are reported in Appendix B.

4.2 Metrics

We evaluate model performance through a hybrid quantitative and qualitative framework.

First, we measure primary task performance using standard **Accuracy (Acc)**, defined as the proportion of instances for which the model selects the gold answer among the N candidates. Second, we report the **Reasoning Retention Rating (R3)** as a secondary measure of global robustness: *how much of a model’s native 10-option performance survives under the evaluated stress conditions?* R3 aggregates baseline-relative accuracy across cardinalities and windows, weighting larger losses more heavily so that catastrophic failures are not masked by easier conditions. Scores range from 0 to 100, with higher values indicating greater robustness. For example, a model with 80% baseline accuracy and $R3=64.5$ retains 64.5% of that performance, corresponding to approximately 51.6% on the original accuracy scale; R3 is therefore not an absolute accuracy score. Appendix C provides the full formulation and shows that an unweighted retention mean produces the same six-model ranking.

Finally, we conduct a **qualitative CoT audit** by parsing the generated reasoning paths.

5 Results

Figure 2 plots model performance across different option cardinalities, window positions, and placement modes, tracking Option-Independent (solid

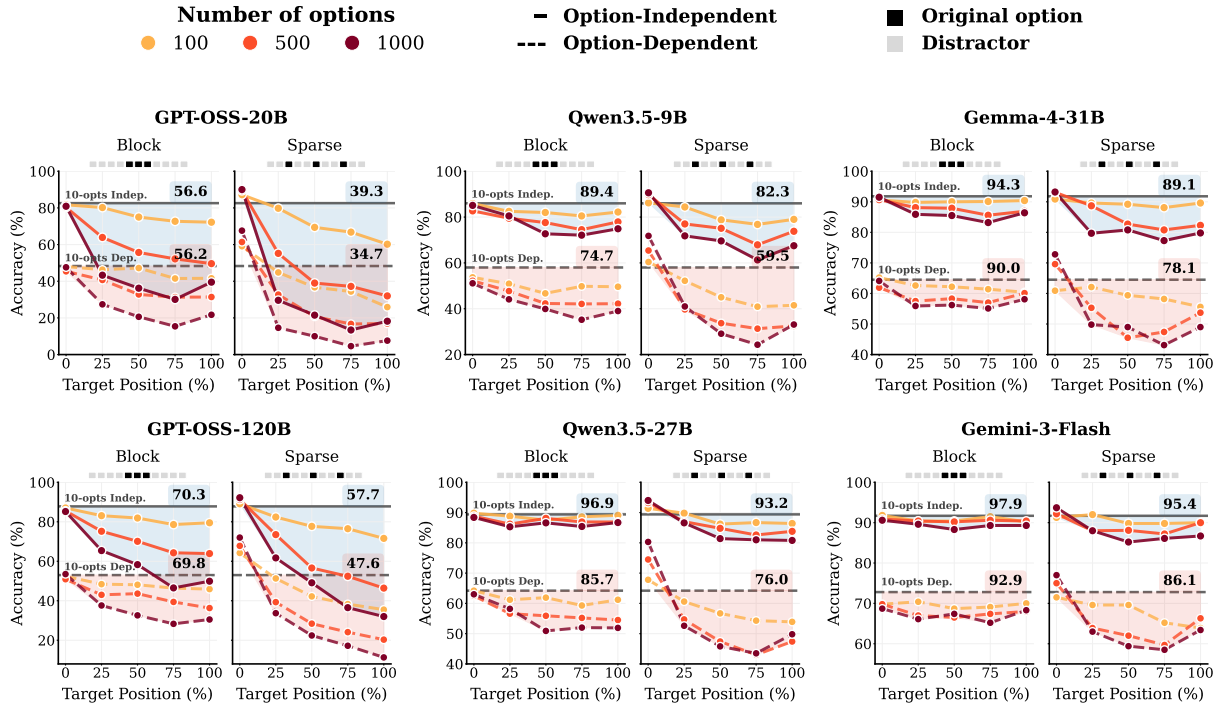


Figure 2: **Lost-in-Choice phenomenon in MMLU-XT.** The original 10 MMLU-Pro options are either contiguous (*Block*) or interleaved with distractors (*Sparse*). *Option-Independent* (solid) is solvable prior to choice inspection; *Option-Dependent* (dashed) requires comparative analysis of the candidates. Dashed gray lines mark the 10-option baselines; shaded labels report R3 across $N \in \{100, 500, 1000\}$.

lines) and Option-Dependent tasks (dashed lines) separately. Tables 2 and 3 synthesize the worst-case degradation curves; complete per-domain trajectories are in Appendix D. Across the 180 combinations of model, option cardinality, window position, and placement mode ($6 \times 3 \times 5 \times 2$), Wilson 95% confidence intervals are narrow (5.6% wide on average; 6.3 maximum), and all main paired contrasts are significant at $p < 10^{-4}$ (Appendix F). Across three distractor seeds, the maximum standard deviation is 2.4 points, roughly 25 times smaller than the largest 60.5-point positional swing (Appendix G).

5.1 Factorial Decomposition of Degradation

The expected decline with N is the aggregate surface of four separable effects: architecture, task dependency, layout, and domain.

Model Size. Model architecture determines the steepness of this decline. Older models (i.e., GPT-OSS-120B) are highly sensitive to scale, whereas modern models (Qwen3.5-27B and Gemma-4-31B) remain more resilient under structured setups.

Question Type. The task nature dictates the penalty severity. For almost all models, the degradation curve is steeper on Option-Dependent questions than on retrieval-driven Option-Independent

Model	Task	10-opt	Block Drop Δ			Sparse Drop Δ		
			100	500	1000	100	500	1000
Gemini-3-Flash	Ind	91.7	1.5	1.5	3.4	1.9	4.5	6.5
	Dep	72.8	4.1	6.3	7.6	8.9	13.1	14.3
Qwen3.5-27B	Ind	89.4	1.9	3.2	4.1	3.2	6.8	8.6
	Dep	64.2	4.9	9.7	13.3	10.3	21.2	20.7
Qwen3.5-9B	Ind	86.0	5.5	11.5	13.9	9.2	18.1	14.2
	Dep	58.0	11.3	26.7	22.8	17.1	26.7	33.8
Gemma-4-31B	Ind	91.8	2.0	6.2	8.6	3.7	11.0	14.5
	Dep	64.5	4.1	7.5	9.4	8.9	19.0	21.4
GPT-OSS-120B	Ind	87.8	9.2	23.9	41.3	10.1	67.5	76.5
	Dep	53.0	7.1	16.7	24.7	17.5	32.7	35.8
GPT-OSS-20B	Ind	82.6	10.4	33.0	52.5	15.8	65.9	78.0
	Dep	48.3	6.8	17.0	32.9	22.4	31.6	38.3

Table 2: **Granular performance degradation curves** (Δ percentage points) across option scales in MMLU-XT. Deltas represent absolute drops from the 10-option baseline for each layout (omitting minus signs for clarity). Cell shading intensity is proportional to Δ ; the largest drop within each Block/Sparse group is bolded.

tasks. A notable anomaly is the GPT-OSS family, which suffers an extreme collapse under independent setups at higher scales.

Layout. Option fragmentation heavily accelerates these losses. Switching from Block to Sparse mode causes additional degradation; just expanding the sparse space to $N = 500$ triggers a severe

Model	Block Drop Δ												Sparse Drop Δ											
	CS			Math			Health			Law			CS			Math			Health			Law		
	100	500	1000	100	500	1000	100	500	1000	100	500	1000	100	500	1000	100	500	1000	100	500	1000	100	500	1000
Gemini-3-Flash	0.8	1.6	2.8	2.0	1.6	3.6	6.0	7.6	8.4	2.8	5.2	5.6	2.0	6.0	8.4	2.4	4.0	4.4	10.4	14.8	16.8	6.4	11.4	10.0
Qwen3.5-27B	1.2	3.6	5.5	0.0	0.6	3.6	9.5	13.4	14.3	3.3	9.1	8.9	2.9	8.6	10.8	1.6	3.1	3.8	15.8	22.6	24.8	7.7	18.7	17.9
Qwen3.5-9B	3.1	9.0	12.3	5.6	12.5	12.9	10.6	15.7	20.9	13.6	18.3	27.3	5.8	17.2	24.6	6.8	13.3	18.7	19.0	24.9	32.6	19.6	32.8	38.3
Gemma-4-31B	2.0	5.3	5.6	1.2	4.8	8.8	5.6	10.7	10.1	6.1	9.7	12.2	2.8	7.8	13.3	1.6	8.5	10.7	8.0	20.0	21.8	11.3	20.8	25.3
GPT-OSS-120B	10.2	25.9	39.0	6.1	25.9	45.4	10.8	18.8	29.1	7.0	13.4	21.2	15.2	42.5	53.3	9.8	37.0	60.8	21.6	35.0	44.8	20.4	35.2	38.4
GPT-OSS-20B	11.2	29.4	45.9	12.0	43.0	74.0	9.2	19.0	31.6	4.4	12.2	22.4	24.4	47.1	69.2	22.0	61.7	81.2	24.8	37.2	48.8	18.4	24.8	31.6

Table 3: **Per-domain performance degradation (Δ absolute percentage points) across option scales in MMLU-XT, grouped by domain.** Deltas represent drops from the 10-option baseline for each layout (omitting minus signs for clarity). Cell shading intensity is proportional to Δ ; the largest drop within each domain/scale block is bolded.

functional collapse, exposing cross-token communication failures (Schnabel et al., 2025).

Domain. Performance resilience scales unevenly across domains. Discursive disciplines (Health, Law) contain more Option-Dependent instances (see Table 1), but composition alone does not explain their penalty. Pooling $N=1000$ over layouts and windows, Option-Dependent accuracy remains lower in Health/Law than in CS/Math even after stratification: GPT-OSS-120B scores 30.4/26.3 vs. 46.9/47.1, and Gemini-3-Flash scores 58.7/62.5 vs. 76.5/79.6. The same ordering already appears at the 10-option baseline (e.g., GPT-OSS-120B: CS 86.4% vs. Law 46.0%), indicating a residual domain-difficulty effect that predates our manipulation. However, switching from Block to Sparse placement systematically amplifies degradation, causing a sharp, linear collapse in Health. Finally, model archetypes diverge under stress: frontier architectures (Gemini, Gemma, Qwen3.5) exhibit small degradation curves, effectively mitigating scaling drops under Block mode. In contrast, Qwen3.5-9B and GPT-OSS show a linear susceptibility to list length; GPT-OSS undergoes a complete attention collapse on formal logic, with Math accuracy dropping up to 81.2% across both layouts.

Global Synthesis via R3 Score The bold, shaded labels situated above the filled curves in Figure 2 report the aggregated R3 score—converting the complex intersecting trajectories of our experimental grid into a standardized, one-glance signature of resilience. The R3 macro-scores establish a clear hierarchy: Gemini-3-Flash, Qwen3.5-27B, and Gemma-4-31B anchor the top tier, indicating greater capability retention under distractor saturation. Beyond individual rankings, R3 reflects the identified macro-trends: it is universally

lower on Option-Dependent tasks than on Option-Independent ones—quantifying the pronounced performance penalty of comparative logic—and it is consistently penalized under Sparse layout.

5.2 Spatial and Positional Bias

Analyzing performance across target positions (see Figure 2) reveals a positional bias that challenges the traditional “lost-in-the-middle” phenomenon.

The degradation trend across context positions is universal: it is active at all option cardinalities (N), simply becoming more severe and visible as N scales up to 1000. Across almost all tested architectures, the performance curves do not exhibit a symmetrical U-shape. Instead of recovering accuracy at the very end of the context, the models suffer their steepest and most destructive drops when the target is located in the late-middle to terminal positions, specifically at window 0.75 and 1.0.

This spatial decay induces a pronounced primacy bias, which becomes particularly acute in Sparse mode. When the fragmented target is placed at the absolute front of the context (window 0.0), almost all configurations—regardless of scale N —show an upward trajectory, often outperforming their original 10-option baseline. One plausible explanation is that early gold evidence anchors processing before downstream distractors accumulate; our behavioral analysis tests the corresponding primacy signature without claiming a specific internal attention mechanism. Finally, this positional vulnerability is consistently sharper on Option-Dependent tasks (dashed lines) than on Option-Independent ones (solid lines). Under high token pressure, an LLM’s capacity to execute multi-option comparative logic survives reasonably well only at the opening of the sequence, collapsing as the target shifts toward the end of the context window.

5.3 The Reasoning vs. Context Paradox

Comparing task types reveals a critical paradox: for all architectures except GPT-OSS, processing 1,000 options for a question whose answer can be derived independently of the candidate set (Option-Independent) is significantly less punishing than resolving a native 10-option question that requires relational reasoning (Option-Dependent).

This choice-structure asymmetry exposes a major blind spot in current LLM evaluation: the bottleneck is not reducible to physical text volume, because the need to compare candidates independently changes resilience. Our findings serve as a warning to the community: long-context benchmarks that rely on retrieval or independent multiple-choice questions overstate architectural maturity, masking a severe performance degradation that occurs when a task requires comparative logic.

5.4 Replication on ToolBench Trajectories

To test whether the positional pattern transfers beyond MMLU-derived questions, we replicate our analysis on ToolBench (Qin et al., 2024), which contains solved agent trajectories involving real APIs. We retain trajectories listing exactly ten tools and use the first tool called in each as the gold answer to a single-step selection task based on the original user request. We then expand each candidate set to 1,000 options by adding 990 real APIs sampled from ToolBench’s full pool and distribute the original ten tools according to the same Sparse layout and window positions used in MMLU-XT. Appendix E gives the full construction details. Qwen3.5-9B and Gemma-4-31B reproduce the same asymmetric late-position collapse on a different task and source distribution (Figure 3). This is evidence that the structural capability failure transfers to real tool-selection trajectories.

5.5 Practical Implications

Large-context selection creates a trade-off with retrieval-based shortlisting. A short list produced through retrieval reduces comparison load but imposes a recall ceiling: a missing target cannot be recovered downstream. Relaxing that filter preserves recall but exposes the model to the cardinality, dispersion, and position effects isolated here. Our results thus support three operational cautions: report performance by target rank rather than only averaged over a retrieved list; preserve locality among mutually relevant candidates where possible; and

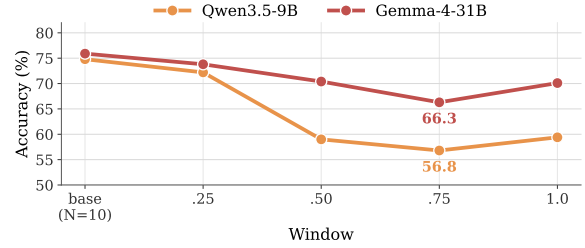


Figure 3: **ToolBench replication accuracy (%)**, 500 real trajectories per cell. Both models remain near baseline at $w=.25$, degrade from $w=.50$, and reach their minimum at $w=.75$, matching MMLU-XT’s trend.

do not infer that a model robust to independent lookup will remain robust when candidates must be compared. Naturally retrieved pools can contain relevance-biased ordering and a hard-negative core followed by weaker items, which may amplify or attenuate the effects observed in our setting.

6 Analysis

Having established that accuracy degrades as option lists grow and shift positionally, we investigate the underlying *mechanisms* driving these failures through four research questions.

6.1 RQ1: What Do LLMs Pick When Wrong?

To determine if scaling options induces confusion, we classify incorrect predictions into two distinct categories: *Wrong Original* (selecting one of the 9 native, plausible distractors) and *Random Distractor* (selecting an injected, less relevant filler option). Figure 4 plots these error trajectories.

Findings. Error composition is not dictated by raw model scale, but by a sharp architectural divide heavily modulated by layout. At $N=100$, random distractor selections are negligible across all setups. Even at maximum scale ($N=1000$), frontier architectures (Gemini-3-Flash, Qwen3.5-27B) maintain tight control over noise, with random picks remaining below 5%. This resilient semantic anchoring is strongly linked to the early-exit effects analyzed in RQ2—models stop scanning once they encounter a plausible original distractor. Conversely, the GPT-OSS family undergoes a complete structural collapse under high context density: their total error rates cross the 70% threshold in fragmented Sparse configurations, with random noise selections spiking dramatically. This severe degradation causes them to underperform the much smaller Qwen3.5-9B, showing that parameter count alone does not determine large-choice resilience.

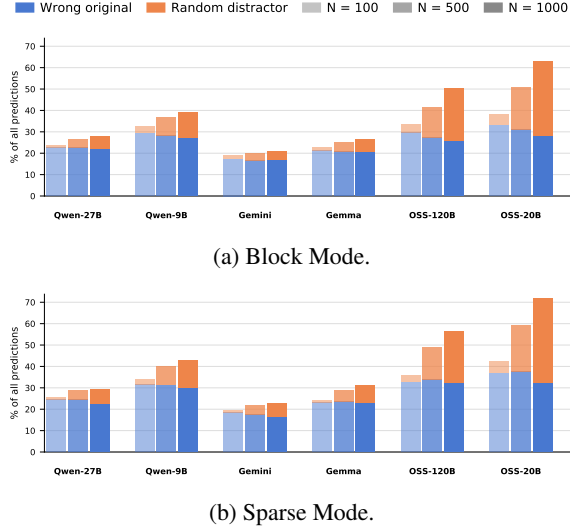


Figure 4: **Wrong-prediction breakdown in MMLU-XT.** Stacked bars: mean rate of wrong-original (blue) and random-distractor (orange) selections as % of all predictions, averaged over window positions. Opacity encodes N : lightest = 100, darkest = 1000.

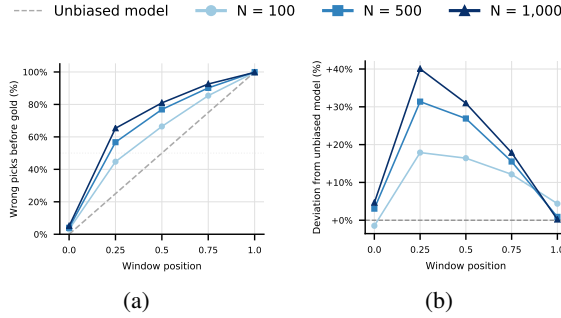


Figure 5: **Positional bias in wrong predictions (Sparse mode).** (a) Before-gold rate vs. unbiased model. (b) Deviation from the unbiased baseline.

6.2 RQ2: Do Models Select Options Appearing Before or After the Gold?

To isolate where models err when making an incorrect prediction, we measure whether the selected distractor appears *before* or *after* the gold answer. Under a position-agnostic baseline, the expected fraction of selections before the gold tracks its relative window position (w). We restrict this analysis exclusively to *Sparse* placement; in Block mode, the contiguous grouping of the nine native distractors would confound option-type preferences with pure spatial bias (see Appendix H). Figure 5b plots the net primacy bias by subtracting the unbiased baseline across all incorrect sparse predictions.

Findings. Spatial distribution analysis reveals a severe, scalable primacy bias that governs model

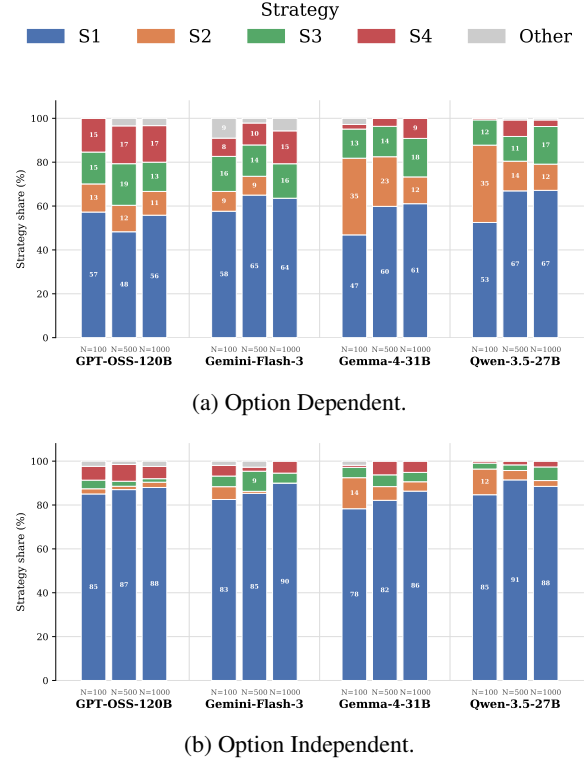


Figure 6: **Strategy distribution by question type and option count.** Each cluster shows three stacked bars ($N=100, 500, 1000$), one per model; Sparse placement, averaged over window positions. Best view if zoomed.

failure modes. The directional preference peaks at intermediate positions ($w \in \{0.25, 0.50\}$), where there is sufficient option mass on both sides of the target for the bias to manifest. Crucially, this effect scales monotonically with context density: at $w=0.25$, the positive deviation from the baseline intensifies from +13% at $N=100$ to +29% at $N=500$, peaking at +39% at $N=1000$. This scaling behavior demonstrates that primacy bias is not a static anchoring to the absolute beginning of the prompt, but rather a proportional pull; longer contexts expand the search space of preceding distractors, amplifying early-exit behaviors and causing models to terminate evaluation prematurely.

6.3 RQ3: What Strategies Do Models Adopt?

To characterize *how* models navigate enlarged option spaces, we first built the S1–S10 taxonomy inductively: we manually reviewed approximately 300 traces stratified across models, option counts, and window positions, iteratively coding recurring behaviors until the scheme stabilized and covered the observed strategies. We then fixed the taxonomy and used an LLM-based classifier to la-

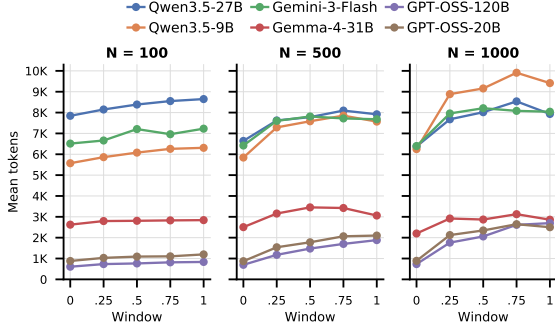


Figure 7: **Scaling of reasoning tokens.** Mean completion tokens per model as a function of window position.

bel the analysis sample (Table 19); the prompt and full sampling protocol are reported in Appendix J.1. We focus on *Sparse* placement—which yields the sharpest degradation and most revealing behavioral signatures—and collapse the taxonomy into four dominant strategies: **S1** (*Independent Derivation then Scan*; analytical solving followed by a target search), **S2** (*Sequential Review*; linear option-by-option evaluation), **S3** (*Eliminative Filtering*; discarding implausible candidates), and **S4** (*Candidate-Led Shortlisting*; matching internally generated hypotheses against the list), while merging rarer behaviors into *Other*.

Findings. Strategy selection is associated more strongly with task dependency than with model scale, as illustrated in Figure 6. On Option-Independent tasks, S1 heavily dominates (78–91%) and remains invariant to N , confirming that models execute independent lookups regardless of context length. Conversely, Option-Dependent tasks force genuine engagement with the option list: while S1 remains dominant (47–67%), relational reasoning demands a diverse tactical repertoire, redistributing significant probability mass to interactive strategies (S2–S4, spanning 8–35%). Architectures that preserve this strategic diversity at higher N also exhibit greater structural resilience, whereas larger drops coincide with rigid, non-interactive scanning.

6.4 RQ4: How Does Reasoning Effort Scale with Options and Position?

We quantify reasoning effort via completion tokens (Figure 7)—the length of the CoT preceding the final answer—and examine how it relates to accuracy (Figure 8) and varies with option scale N and window position w (Appendix Figure 13).

Findings. Token allocation reveals two architectural regimes: *Heavy Reasoners* (Qwen3.5-9B/27B, Gemini-3-Flash), which use 6K–10K tokens and

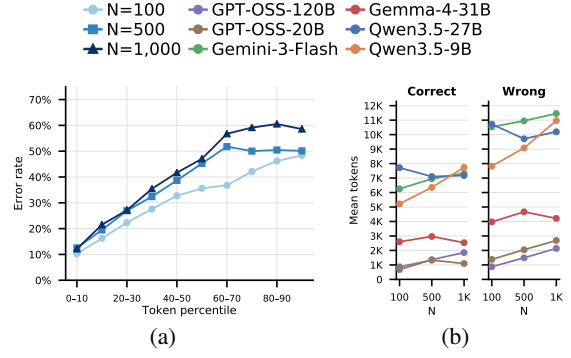


Figure 8: **Reasoning effort correlation with wrong answers.** (a) Error rate per within-model completion-token decile, pooled across all models and N values. (b) Mean completion tokens on correct vs. incorrect answers per model and N .

expand their budgets as N grows, and *Light Reasoners* (GPT-OSS, Gemma-4-31B), which remain below 3K tokens. Across both families, token usage increases with both option pool size (N) and gold target depth (w). Crucially, within-model decile analysis (Figure 8a) shows error rates rising from ~10% in the lowest token decile to ~60% in the highest. Incorrect traces are consistently longer than correct ones (Figure 8b), with frontier models spending an additional 3–4K tokens on failed tasks. Statistical analyses further confirm a robust association between longer reasoning traces and lower correctness, controlling for question difficulty and experimental conditions (Appendix F). This remains observational evidence: longer traces may reflect unresolved search or other failure processes rather than cause errors.

7 Conclusion

Lost in Choice is a structured failure of large-set decision-making. Across six reasoning LLMs, our factorial design shows that option cardinality, layout, and position interact. The Reasoning vs. Context Paradox further shows that models can find relational-reasoning tasks with 10 options more difficult than independently solvable questions with 1,000 options. Large-choice evaluations should therefore report what must be compared, how candidates are arranged, where the target occurs, and how selection strategies change. Longer contexts make more choices visible, but do not ensure that models can choose among them reliably.

Limitations

Our study has several limitations that should be considered when interpreting the results.

First, the main factorial experiment uses multiple-choice questions derived from MMLU-Pro. The ToolBench replication broadens the evidence to real solved tool-selection trajectories, but its 1,000-item candidate expansions remain controlled and exclude the 1.7% of queries with ambiguous near-duplicate targets. Neither experiment estimates how often deployed systems expose a target at a particular rank, and relevance-biased retrieval or reranking may compound or attenuate the observed positional effect. Consequently, our results do not establish the safety, reliability, or prevalence of these failures in deployed systems, particularly across other languages, domain-specific pipelines, or human-in-the-loop workflows.

Second, the injected in-domain distractors are intentionally random. Their exchangeability supports clean contrasts across layout and position, but they do not reproduce every naturally retrieved candidate distribution and make nominal N larger than the unknown effective set of genuinely competitive alternatives. Our effect magnitudes should thus be interpreted as estimates for this controlled distribution, not as lower or upper bounds for deployment. A complementary study should construct hardness-matched pools, repeat candidate permutations, and model relevance ordering explicitly.

Third, all six evaluated systems are reasoning models under a minimal zero-shot prompt. Scale-aware instructions, two-stage shortlisting, tree search, and dynamically loaded candidate sets may mitigate the failure, but comparing such interventions at the full 31,000-instance scale is outside the scope of this diagnostic study.

Societal Impact and Ethical Considerations

The findings have implications for the use of LLMs in systems that assist or automate decisions. The observed *Lost in Choice* effect suggests that increasing the number of available alternatives does not make a decision system more capable; instead, it can introduce systematic changes in which candidates are considered and ultimately selected. This is particularly relevant when choices concern consequential domains such as healthcare, law, or access to services, where minor changes in candidate ordering or presentation could influence outcomes.

A key ethical concern is that these failures are

not necessarily visible from aggregate accuracy alone. Models may appear competent on conventional benchmarks while exhibiting substantially different behavior when required to compare many alternatives. This creates a risk of misplaced trust if benchmark performance is interpreted as evidence of reliable decision-making. Evaluation and deployment should therefore consider not only whether a model produces the correct answer, but also whether its decisions remain stable under changes in candidate-set structure.

The results also motivate greater emphasis on human oversight in high-stakes applications. When an LLM is used to rank, filter, or select among many candidates, its output should not be treated as an inherently neutral ordering of the available alternatives. In particular, system designers should avoid allowing model-generated selections to become opaque gatekeeping mechanisms without appropriate review. More broadly, the study highlights the need to treat candidate-set design as part of the decision process itself: how alternatives are presented can affect the outcome, even when the underlying information remains unchanged.

Acknowledgements

Research partially supported by AI-PACT project (CUP B47H22004450008, B47H22004460001); National Plan PNC-I.1 DARE initiative (PNC0000002, CUP B53C22006450001); PNRR Extended Partnership FAIR (PE00000013, Spoke 8); 2024 Scientific Research and High Technology Program, project “AI analysis for risk assessment of empty lymph nodes in endometrial cancer surgery”, the Fondazione Cassa di Risparmio in Bologna; Chips JU TRISTAN project (G.A. 101095947).

References

- Nishant Balepur, Rachel Rudinger, and Jordan Lee Boyd-Graber. 2025. [Which of these best describes multiple choice evaluation with LLMs? A\) forced B\) flawed C\) fixable D\) all of the above](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2025, Vienna, Austria, July 27 - August 1, 2025, pages 3394–3418. Association for Computational Linguistics.
- Alessio Cocchieri, Luca Ragazzi, Giuseppe Tagliavini, and Gianluca Moro. 2026a. [LLMs \(almost\) never abstain under medical uncertainty](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2026, San Diego, California, United States, July 2-7, 2026, pages 29573–29613. Association for Computational Linguistics.
- Alessio Cocchieri, Luca Ragazzi, Giuseppe Tagliavini, and Gianluca Moro. 2026b. [ReMedQA: Are we done with medical multiple-choice benchmarks?](#) In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2706–2738, Rabat, Morocco. Association for Computational Linguistics.
- Pietro di Lena, Giacomo Domeniconi, Luciano Margara, and Gianluca Moro. 2015. [GOTA: GO term annotation of biomedical literature](#). *BMC Bioinform.*, 16:346:1–346:13.
- Giacomo Domeniconi, Gianluca Moro, Andrea Pagliarani, and Roberto Pasolini. 2015. [Markov chain based method for in-domain and cross-domain sentiment classification](#). In *KDIR 2015 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*, part of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015), Volume 1, Lisbon, Portugal, November 12-14, 2015, pages 127–137. SciTePress.
- Giacomo Domeniconi, Gianluca Moro, Andrea Pagliarani, and Roberto Pasolini. 2017. [On deep learning in cross-domain sentiment classification](#). In *Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - (Volume 1)*, Funchal, Madeira, Portugal, November 1-3, 2017, pages 50–60. SciTePress.
- Giacomo Domeniconi, Gianluca Moro, Roberto Pasolini, and Claudio Sartori. 2014. [Iterative refining of category profiles for nearest centroid cross-domain text classification](#). In *Knowledge Discovery, Knowledge Engineering and Knowledge Management - 6th International Joint Conference, IC3K 2014, Rome, Italy, October 21-24, 2014, Revised Selected Papers*, volume 553 of *Communications in Computer and Information Science*, pages 50–67. Springer.
- Yu Fan, Jingwei Ni, Jakob Merane, Etienne Salimbeni, Yang Tian, Yoan Hermstrüwer, Yinya Huang, Mubashara Akhtar, Florian Geering, Oliver Dreyer, Daniel Brunner, Markus Leippold, Mrinmaya Sachan, Alexander Stremitzer, Christoph Engel, Eliott Ash, and Joel Niklaus. 2025. [LEXam: Benchmarking legal reasoning on 340 law exams](#). *CoRR*, abs/2505.12864.
- Ilaria Ferrari, Giacomo Frisoni, Paolo Italiani, Gianluca Moro, and Claudio Sartori. 2022. [Comprehensive analysis of knowledge graph embedding techniques benchmarked on link prediction](#). *Electronics*, 11(23).
- Giacomo Frisoni, Alessio Cocchieri, Alex Presepi, Gianluca Moro, and Zaiqiao Meng. 2024. [To generate or to retrieve? On the effectiveness of artificial contexts for medical open-domain question answering](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9878–9919, Bangkok, Thailand. Association for Computational Linguistics.
- Giacomo Frisoni and Gianluca Moro. 2020. [Phenomena explanation from text: Unsupervised learning of interpretable and statistically significant knowledge](#). In *Data Management Technologies and Applications - 9th International Conference, DATA 2020, Virtual Event, July 7-9, 2020, Revised Selected Papers*, volume 1446 of *Communications in Computer and Information Science*, pages 293–318. Springer.
- Gemma Team. 2026. [Gemma 4: Byte for byte, the most capable open models](#).
- Daya Guo, Dejian Yang, Haowei Zhang, and Junxiao Song et al. 2025. [DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning](#). *Nature*, 645(8081):633–638.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. [Measuring massive multitask language understanding](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Paolo Italiani, Luca Ragazzi, and Gianluca Moro. 2025. [Read between the tokens: Differentiable text pruning via perturbed top-k selection](#). *IEEE Transactions on Audio, Speech and Language Processing*, 33:4870–4879.
- Sheena S Iyengar and Mark R Lepper. 2000. When choice is demotivating: Can one desire too much of a good thing? *Journal of personality and social psychology*, 79(6):995.
- Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. [What disease does this patient have? A large-scale open domain question answering dataset from medical exams](#). *CoRR*, abs/2009.13081.
- Nahyun Lee and Guijin Son. 2026. [Pushing the boundaries of multiple choice evaluation to one hundred options](#). *CoRR*, abs/2604.14634.

- Ruizhe Li and Yanjun Gao. 2025. [Anchored answers: Unravelling positional bias in GPT-2’s multiple-choice questions](#). In [Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025, Findings of ACL](#), pages 2439–2465. Association for Computational Linguistics.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. [Lost in the middle: How language models use long contexts](#). [Trans. Assoc. Comput. Linguistics](#), 12:157–173.
- Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. [Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity](#). In [Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics \(Volume 1: Long Papers\)](#), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 8086–8098. Association for Computational Linguistics.
- Lorenzo Molfetta, Alessio Cocchieri, Stefano Fantazzini, Giacomo Frisoni, Luca Ragazzi, and Gianluca Moro. 2025. [FEAST: Retrieval-augmented multi-hierarchical food classification for the FoodEx2 system](#). In [ECAI 2025 - 28th European Conference on Artificial Intelligence, 25-30 October 2025, Bologna, Italy - Including 14th Conference on Prestigious Applications of Intelligent Systems \(PAIS 2025\)](#), volume 413 of [Frontiers in Artificial Intelligence and Applications](#), pages 4169–4176. IOS Press.
- Lorenzo Molfetta, Alessio Cocchieri, Luca Ragazzi, Ilaria Bartolini, Marco Patella, and Gianluca Moro. 2026. [Sycophants in the courtroom: Are LLMs fragile to juridical authority and evolving legal standards?](#) In [Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics \(Volume 1: Long Papers\)](#), ACL 2026, San Diego, California, United States, July 2-7, 2026, pages 10865–10886. Association for Computational Linguistics.
- Gianluca Moro and Luca Ragazzi. 2022. [Semantic self-segmentation for abstractive summarization of long documents in low-resource regimes](#). In [Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022](#), pages 11085–11093. AAAI Press.
- Gianluca Moro and Luca Ragazzi. 2023. [Align-then-abstract representation learning for low-resource summarization](#). [Neurocomputing](#), 548:126356.
- Gianluca Moro, Luca Ragazzi, and Lorenzo Valgimigli. 2023. [Graph-based abstractive summarization of extracted essential knowledge for low-resource scenarios](#). In [ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland - Including 12th Conference on Prestigious Applications of Intelligent Systems \(PAIS 2023\)](#), volume 372 of [Frontiers in Artificial Intelligence and Applications](#), pages 1747–1754. IOS Press.
- Bennet B Murdock Jr. 1962. The serial position effect of free recall. [Journal of experimental psychology](#), 64(5):482.
- OpenAI. 2023. [GPT-4 technical report](#). [CoRR](#), abs/2303.08774.
- OpenAI. 2025. [gpt-oss-120b & gpt-oss-20b model card](#). [CoRR](#), abs/2508.10925.
- Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2024. [YaRN: Efficient context window extension of large language models](#). In [The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024](#). OpenReview.net.
- Pouya Pezeshkpour and Estevam Hruschka. 2024. [Large language models sensitivity to the order of options in multiple-choice questions](#). In [Findings of the Association for Computational Linguistics: NAACL 2024](#), pages 2006–2017, Mexico City, Mexico. Association for Computational Linguistics.
- Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. [ToolLLM: Facilitating large language models to master 16000+ real-world APIs](#). In [The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024](#). OpenReview.net.
- Qwen Team. 2026. [Qwen3.5: Towards native multi-modal agents](#).
- Luca Ragazzi, Paolo Italiani, Gianluca Moro, and Mattia Panni. 2024. [What are you token about? Differentiable perturbed top-k token selection for scientific document summarization](#). In [Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024, volume ACL 2024 of Findings of ACL](#), pages 9427–9440. Association for Computational Linguistics.
- Luca Ragazzi, Gianluca Moro, Lorenzo Valgimigli, and Riccardo Fiorani. 2025. [Cross-document distillation via graph-based summarization of extracted essential knowledge](#). [IEEE Transactions on Audio, Speech and Language Processing](#), 33:518–527.
- Leonardo Ranaldi and Giulia Pucci. 2023. [When large language models contradict humans? large language models’ sycophantic behaviour](#). [CoRR](#), abs/2311.09410.

- Mathieu Ravaut, Aixin Sun, Nancy F. Chen, and Shafiq Joty. 2024. [On context utilization in summarization with large language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2024, Bangkok, Thailand, August 11-16, 2024, pages 2764–2781. Association for Computational Linguistics.
- Tobias Schnabel, Kiran Tomlinson, Adith Swaminathan, and Jennifer Neville. 2025. [Lost in transmission: When and why LLMs fail to reason globally](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*.
- Zhengliang Shi, Yuhan Wang, Lingyong Yan, Pengjie Ren, Shuaiqiang Wang, Dawei Yin, and Zhaochun Ren. 2025. [Retrieval models aren’t tool-savvy: Benchmarking tool retrieval for large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24497–24524, Vienna, Austria. Association for Computational Linguistics.
- Raphael Tang, Crystina Zhang, Xueguang Ma, Jimmy Lin, and Ferhan Ture. 2024. [Found in the middle: Permutation self-consistency improves listwise ranking in large language models](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2327–2340, Mexico City, Mexico. Association for Computational Linguistics.
- Runchu Tian, Yanghao Li, Yuepeng Fu, Siyang Deng, Qinyu Luo, Cheng Qian, Shuo Wang, Xin Cong, Zhong Zhang, Yesai Wu, Yankai Lin, Huadong Wang, and Xiaojiang Liu. 2025. [Distance between relevant information pieces causes bias in long-context LLMs](#). In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025, Findings of ACL*, pages 521–533. Association for Computational Linguistics.
- Haochun Wang, Sendong Zhao, Zewen Qiang, Nuwa Xi, Bing Qin, and Ting Liu. 2025. [LLMs may perform MCQA by selecting the least incorrect option](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5852–5862, Abu Dhabi, UAE. Association for Computational Linguistics.
- 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 Wenhui Chen. 2024. [MMLU-Pro: A more robust and challenging multi-task language understanding benchmark](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. [Chain of thought prompting elicits reasoning in large language models](#). *ArXiv*, abs/2201.11903.
- Zhen Yang, Ping Jian, and Chengzhi Li. 2025. [Option symbol matters: Investigating and mitigating multiple-choice option symbol bias of large language models](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1902–1917, Albuquerque, New Mexico. Association for Computational Linguistics.
- Guangxiang Zhao, Saier Hu, Xiaoqi Jian, Jinzhu Wu, Yuhan Wu, Lin Sun, and Xiangzheng Zhang. 2025. [Large language models badly generalize across option length, problem types, and irrelevant noun replacements](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 26837–26846. Association for Computational Linguistics.
- Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. [Large language models are not robust multiple choice selectors](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Yuxin Zuo, Shang Qu, Yifei Li, Zhang-Ren Chen, Xuekai Zhu, Ermo Hua, Kaiyan Zhang, Ning Ding, and Bowen Zhou. 2025. [MedXpertQA: Benchmarking expert-level medical reasoning and understanding](#). In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025, volume 267 of Proceedings of Machine Learning Research*. PMLR / OpenReview.net.

A Data Preprocessing

A.1 Question Classification

Each of the 2,858 MMLU-Pro questions drawn from the four target domains (Computer Science, Mathematics, Law, Health) is labelled as either Option-Dependent or Option-Independent by an ensemble of three LLM judges: GPT-OSS-120B, Gemma-4-31B, and Qwen3.5-9B. A question is Option-Dependent if the correct answer cannot be determined without inspecting the option list (e.g., “which of the following is not true?”, numerical best-approximation items, or questions whose answer is defined relative to the set of alternatives); it is Option-Independent if the answer can be derived from the question stem alone, independently of what options are presented.

Each judge is queried independently via the Together AI Batch API using the same zero-shot classification prompt (Figure 14). Final labels are assigned by majority vote across the three judges.

To account for domain variation known to affect classification (Domeniconi et al., 2014, 2015, 2017), two authors independently annotated a subset of 200 questions (50 per domain), achieving Cohen’s $\kappa = 0.87$. Agreement between the ensemble’s majority-vote labels and the human reference labels averaged $\kappa = 0.78$ across domains (CS = 0.82, Health = 0.80, Law = 0.76, Math = 0.74).

Table 5 provides representative examples from all four MMLU-XT domains.

A.2 Distractor Quality Validation

To ensure that performance degradation is strictly attributable to cardinality rather than label noise or semantic ambiguity, we conduct a systematic validation of the $N = 1000$ distractor pool. We define *coincidental correctness* as any distractor d_i that could be considered a defensible or partially correct answer under expert scrutiny.

Judge-Based Screening We evaluated 990,000 question-distractor pairs using a rigorous dual-judge protocol: GPT-OSS-120B for general domains and Gemini-3-Flash for Mathematics, selected for its superior precision in numerical verification and algebraic equivalence detection. Judges were prompted to flag any distractor exhibiting semantic overlap or logical consistency with the ground truth (Figure 15). As shown in Table 4, 51.5% of the questions contained at least one flagged distractor. The Law domain exhib-

Domain	Flagged Qs (%)	Avg/Q	Avg/FQ	Dup%
CS	107 / 250 (42.8%)	0.96	2.24	1.7%
Math	128 / 250 (51.2%)	5.79	11.30	6.4%
Law	153 / 250 (61.2%)	2.24	3.66	1.0%
Health	127 / 250 (50.8%)	1.61	3.17	1.0%
Overall	515 / 1000 (51.5%)	2.65	5.14	2.5%

Table 4: **Distractor validation and correction statistics.** *Flagged Qs*: questions with ≥ 1 flagged distractor. *Avg/Q*: mean flags per question (all). *Avg/FQ*: mean flags per flagged questions. *Dup%*: percentage of judge-suggested replacements that resulted in duplicates.

ited the highest proportion of flagged questions (61.2%), consistent with the tendency of dense legal text to generate partial semantic overlaps. Conversely, Mathematics showed the highest flag density (avg = 11.30 per flagged question), reflecting the high prevalence of numerical collisions inherent to high-cardinality sampling.

To bound automated judge errors, two authors independently audited a randomized sample of 200 question-distractor pairs (100 flagged, 100 passed), achieving high consensus (Cohen’s $\kappa = 0.82$). The audit revealed a 94.0% judge precision in the nuanced Law domain and a global false negative rate $< 4.5\%$. This empirical validation effectively bounds label noise, confirming that performance drops are driven by experimental geometry rather than screening artifacts.

Correction and Fallback Protocol All flagged distractors were replaced via a two-tier pipeline:

- **Direct Replacement (84%)**: We applied a judge-generated alternative to be incorrect yet lexically and contextually plausible.
- **Safe Fallback (16%)**: When suggestions collided with existing choices, we sampled a unique replacement from the verified-safe pool $\mathcal{D}_{\text{safe}} = \{d \in \text{Pool} \mid \text{flag}(d) = \text{False} \wedge d \notin \text{Originals} \cup \text{Used}\}$, where *Used* tracks replacements already assigned to the current instance to strictly prevent duplicate entries.

This protocol guarantees that the gold answer remains structurally protected; we formally verified that zero replacement candidates matched the gold label across the entire 31,000-instance suite.

Structural Invariance Corrections were deterministically derived from the reference $N = 1000$ pool (Block mode, $w = 0$) and propagated across all smaller cardinality scales (N_{100} , N_{500}), placement

Domain	Example (truncated)	Rationale
Option-Dependent		
Health	Q6473. An increased frequency of urination in older adults is most likely caused by H. Smaller bladder (gold) B. Bladder infection [... 10 options total]	“Most likely” forces a comparison of relative likelihood across several plausible causes (infection, diabetes, diuretics, smaller bladder); no single cause is uniquely implied by the stem alone.
Law	Q1010. A man and a 14-year-old girl were arrested for smoking marijuana together; he was charged with possession and contributing to the delinquency of a minor, and she was charged as an accomplice to that same offense. At trial, the girl’s best defense is that E. since the statute was designed to protect minors, she cannot be prosecuted as an accomplice (gold) H. a minor cannot be prosecuted for an adult crime [... 10 options total]	Several distractors state legally plausible-sounding defenses; the correct doctrine (a statute’s protected class cannot be an accomplice to a crime defined to protect that class) is only identifiable by weighing it against the other framings.
Math	Q8015. Which is the best buy? D. 20 bars of soap for \$45.00 (gold) B. 16 bars of soap for \$36.48 [... 10 options total]	“Best buy” is inherently comparative: each option’s unit price must be computed and ranked against all others – there is no best-buy fact independent of the option set.
CS	Q10560. Statement 1: The BLEU metric uses precision, while the ROUGE metric uses recall. Statement 2: Hidden Markov models were frequently used to model English sentences. F. True, True (gold) H. False, False [... 10 options total]	Options combine judgments on two separate statements into composite labels (True/False/Mostly True/Cannot be determined); the evaluation space is defined by the options, not the stem, matching the rubric’s “Not Given”-style dependency rule.
Option-Independent		
Health	Q6344. The pattern of inheritance shown by glucose-6-phosphate dehydrogenase (G6PD) deficiency is: E. X-linked recessive (gold) A. Y-linked recessive [... 10 options total]	A fixed genetics fact about a named condition – recallable directly without needing to see or compare any options.
Law	Q1411. In the absence of a provision to the contrary in the articles of incorporation, the directors of a corporation elected for a specified term E. can be removed from office at a meeting of the shareholders, with or without cause (gold) A. can be removed from office at a meeting of the shareholders, but only with a two-thirds majority vote [... 10 options total]	A specific default corporate-law rule with one correct statement of the doctrine, independently recallable from legal knowledge.
Math	Q8730. In how many ways can we color a loop of 5 vertices with 3 colors such that no two adjacent vertices have the same color? C. 30 (gold) A. 60 [... 10 options total]	A closed-form combinatorial computation (chromatic polynomial of a 5-cycle) yields one specific numeric answer, computed independently of the option list.
CS	Q10639. Failed sessions allow brute-force attacks on access credentials. This type of attack is carried out at which layer of the OSI model? J. Session layer (gold) A. Data-link layer [... 10 options total]	A single named technical fact (session-layer responsibility for session/credential handling) with one independently verifiable correct answer.

Table 5: **Qualitative examples of option-dependent and option-independent questions across four domains.** Samples are from questions where all three categorization judges (Gemma-4-31B, GPT-OSS-120B, Qwen3.5-9B) unanimously agreed on the label. Options are truncated to the gold answer and one contrasting distractor for space; each example has 10 options in the underlying benchmark.

Domain	Mean words	Median words
Mathematics	3.6	1
Health	5.0	4
Computer Science	5.8	3
Law	13.9	13

Table 6: **Length of original answer options by domain.** Law’s maximum is 101 words.

modes, and window positions via exact text matching. This process is mathematically sound because the distractor pool remains invariant across spatial configurations for a given seed, and smaller scales are constructed as strict prefixes of the $N = 1000$ setup. This invariance ensures that observed variations in model performance are driven by experimental geometry and cardinality rather than changing distractor content. Examples of flagged distractors and corrections are provided in Table 18.

Option Length as a Potential Confound Table 6 shows that Law has by far the longest original options, whereas Math has the shortest. Yet for GPT-OSS-20B at $N=1000$, Math drops by 74.0/81.2 percentage points in Block/Sparse mode, compared with 22.4/31.6 for Law. Thus longer option text does not monotonically predict larger degradation. More decisively, at fixed N our window sweep reuses the identical multiset of texts, making total prompt length essentially invariant while accuracy varies by as much as 60.5 points across positions.

B Experimental Setup

Prompt Template Each instance is presented as a zero-shot user message (no system prompt) with the following structure:

```
Question: <question text>
Options:
(0) <option 0>
(1) <option 1>
...
(N-1) <option N-1>
Select the single correct option.
Respond with JSON: {"answer_index":
<integer from 0 to N-1>}
```

Models We evaluate six LLMs spanning a range of sizes and providers. Table 11 lists each check-point, access method, and model page.

Structured Output All requests set the response format to a strict JSON schema enforcing `{"answer_index": integer}`, with the integer constrained to $[0, N-1]$. This eliminates letter-to-index conversion and parsing failures entirely: the

model is forced to emit a valid index as its final response, making evaluation deterministic.

Reasoning Effort All six models emit a chain-of-thought trace enclosed in `<think>...</think>` tags before the structured response; the trace is preserved for strategy analysis (Section 6.3) but plays no role in answer extraction. Reasoning budgets are configured per model family:

- **Qwen3.5 (9B, 27B) and Gemma-4-31B.** These models expose no reasoning-budget parameter, so we set `max_tokens = 32,000` to avoid truncating long chains at large N .
- **GPT-OSS (20B, 120B).** We set a medium reasoning effort. Preliminary runs at high effort revealed severe instability: circular reasoning loops, truncated traces, and ill-formed JSON outputs yielded an invalidity rate of $\sim 25\%$ of samples, making higher settings unsuitable for reliable evaluation.
- **Gemini-3-Flash.** We set an high thinking level to grant the model its full reasoning capacity.

Decoding Configuration All models are evaluated under default decoding settings. For open-source models, we follow the parameters specified in the `generation_config.json` of the respective Hugging Face repository; where no configuration file is present, we adopt the defaults recommended in the model’s README. For Gemini-3-Flash, we use the Vertex AI API defaults.

Batch Execution Open-source models are processed via the Together AI Batch API; Gemini-3-Flash via the Vertex AI Batch API. The sole exception is Qwen3.5-27B, which is served locally using vLLM. Qwen3.5 models and Gemma-4-31B are executed in FP8 precision. Each submitted job covers one (model, window, placement mode, seed) combination, corresponding to 1,000 questions.

C R3 Formulation

To penalize localized vulnerabilities that traditional macro-averages inherently mask, R3 implements a two-level dynamic weighting scheme:

1. **Within-Window Resilience:** For a given window w , we measure performance degradation across option scales $N \in \{100, 500, 1000\}$ relative to baseline performance Acc_{10} . To ensure accidental performance gains do not mask

Model	R3	Unweighted
Gemini-3-Flash	93.1	94.8
Qwen3.5-27B	88.0	91.8
Gemma-4-31B	87.9	91.6
Qwen3.5-9B	76.5	82.7
GPT-OSS-120B	61.4	75.5
GPT-OSS-20B	46.7	66.8

Table 7: **R3 sensitivity analysis.** Both specifications induce the same ordering.

losses, the drop $D_{N,w}$ is bounded below by zero:

$$D_{N,w} = \max(0, \text{Acc}_{10} - \text{Acc}(N, w)) \quad (1)$$

We formalize the dynamic scale-weight $W_{N,w}$ as a function of the drop magnitude: $W_{N,w} = \exp(D_{N,w})$. This exponential formulation ensures that severe, catastrophic degradations strictly dominate the window’s final score, while minor, linear decays are downweighted. To prevent numerical instability when a model exhibits near-zero baseline performance, we introduce a smoothing stabilization factor $\epsilon = 10^{-5}$. The localized resilience percentage R_w is:

$$R_w = \frac{\sum_N W_{N,w} \cdot \left(1 - \frac{D_{N,w}}{\text{Acc}_{10} + \epsilon}\right)}{\sum_N W_{N,w}} \times 100 \quad (2)$$

where $R_w = 100$ if $\sum_N D_{N,w} = 0$, and $R_w = 0$ if $\text{Acc}_{10} < \epsilon$.

2. **Cross-Window:** Symmetrically, we project local scores into decimal losses: $L_w = (100 - R_w)/100$. Window importance is scaled dynamically by its relative damage, $W_w = L_w / (\max_{w'} L_{w'} + \epsilon)$, yielding the R3 score:

$$R3 = \left(1 - \frac{\sum_w (L_w \cdot W_w)}{\sum_w W_w + \epsilon}\right) \times 100 \quad (3)$$

Sensitivity to an Unweighted Alternative We replace both dynamic weighting stages with the simple mean of $\min(\text{Acc}(N, w), \text{Acc}_{10}) / \text{Acc}_{10}$ over the same 15 (N, w) cells, then average over layouts and dependency classes. Although the unweighted score is systematically less punitive, it recovers the identical model ranking (Table 7).

D Per-Model Detailed Results

Tables 12–17 report accuracy broken down by placement mode, option count N , and window position, with columns covering overall accuracy, domain slices (CS, Health, Law, Math), option-dependency class, and error type (original wrong options vs. random distractors).

MMLU-XT component	ToolBench counterpart
10 original options	the 10 real tools listed in one ToolBench conversation’s system prompt
gold option	the tool the real agent actually called <u>first</u> in a genuine solved trajectory
wrong original options	the other 9 real tools from that same conversation (genuinely task-relevant, just not called first)
random distractors	real APIs pooled from the rest of the parsed ToolBench corpus
Sparse placement	identical placement procedure as used for MMLU-XT, applied unmodified
window sweep	identical formula, identical positions $w \in \{0, .25, .5, .75, 1\}$

Table 8: Mapping from MMLU-XT benchmark components onto the ToolBench replication.

E ToolBench Replication: Construction Details

We give the full construction pipeline behind Section 5.4: the correspondence between each MMLU-XT benchmark component and its ToolBench counterpart (Table 8), the data-quality filters applied, the evaluation prompt, and a concrete worked example.

Source Data and Trajectory Parsing Trajectories are drawn from the training split of ToolBench²—real system/user/assistant/function conversations produced by ToolBench’s depth-first search-based decision tree (DFSdT) solver. For each conversation, we parse the tool pool from the system prompt’s API list and take gold as the function name on the first assistant turn’s action.

Why “First Tool Called” As Gold ToolBench’s own evaluation protocol (ToolEval) does not use fixed gold labels: the great majority of instructions require multiple sequential tool calls, so no single tool fully resolves them, and the original paper instead scores full trajectories with an LLM judge. Our framework requires one fixed gold index per instance to sweep window position, which trajectory-level judging does not provide. “Which tool would you call first” is well-posed for every retained trajectory and is answered directly from the recorded solved run, not by annotation.

²<https://huggingface.co/datasets/tuandunghcmut/toolbench-v1>

Filtering We require (i) exactly ten originally-listed tools per query, matching MMLU-XT’s fixed $N_{\text{orig}} = 10$, and (ii) globally unique instruction text across the raw corpus, since ToolBench recycles generic instruction text across unrelated tool contexts (e.g., the same query text paired with tool sets that share no plausible relevance, which we verified manually before excluding). These two filters leave 2,161 queries. Of these, 36 (1.7%) have a gold tool with a provider- or version-suffixed sibling among the other nine originally-listed tools—e.g., gold provider4_live_inplaying alongside siblings provider2_live_inplaying and provider3_live_inplaying—detected via regex normalization of provider numerals and version suffixes. DFSDT’s choice among such siblings is arbitrary, so we exclude these 36, leaving 2,125 queries from which every cell samples 500.

Distractor Pool Random distractors are drawn from a pool of 11,445 unique real APIs collected across the full parsed corpus (the union of every conversation’s system-prompt tool list).

Evaluation Prompt The following template is used verbatim for every instance, with <query> and the numbered candidate list substituted in; N is the current option count.

You are given a user request and a numbered list of candidate tools/APIs. The request may need several tool calls in sequence to fully complete. Select ONLY the FIRST tool/API you would call to begin handling this request – not one that alone finishes the whole request.
User request: <query>
Candidate tools:
(0) <tool 0>
...
(N-1) <tool N-1>
Respond with JSON: {"answer_index": <integer from 0 to N-1>} – the index of the FIRST tool you would call.

Worked Example A real retained query is presented below. tiktok_private::get_music—the tool the real trajectory called first, even though the request also implies music_posts and end_a_live_stream would be needed later. When scaled to $N=1000$ options, the other 9 real originally-listed tools are scattered elsewhere in the list among the distractors, under Sparse placement.

User request: “I’m a huge fan of TikTok and I’m interested in exploring posts related to a specific music. Can you fetch the posts associated with the music identified by the ID ‘12345’? Additionally,

Model	b/c	McNemar p
GPT-OSS-120B	500/8	2.5×10^{-136}
GPT-OSS-20B	579/5	1.8×10^{-164}
Qwen3.5-27B	162/29	1.3×10^{-23}
Qwen3.5-9B	300/19	3.6×10^{-66}
Gemma-4-31B	189/19	2.1×10^{-36}
Gemini-3-Flash	131/34	1.2×10^{-14}

Table 9: **Paired 10-option baseline vs. worst $N=1000$ cell.** b/c counts correct→incorrect / incorrect→correct switches.

provide me with the details of the music with the same ID. Furthermore, I need to end a live stream with the stream ID ‘9876’ and room ID ‘5432’.”

The 10 originally-listed tools:
valid_whatsapp::check_phone_number,
scrapitik::search_posts,
scrapitik::end_a_live_stream,
scrapitik::video_without_watermark,
onlyfans::sign_info, onlyfans::count,
onlyfans::performer_model_info,
tiktok_private::username_to_user_id,
tiktok_private::get_music (gold),
tiktok_private::music_posts.

F Statistical Inference

For each of the 180 model- N -window-layout cells (up to 1,000 questions), Wilson 95% confidence intervals have an average width of 5.6 percentage points and a maximum width of 6.3 points. For paired comparisons, we use McNemar’s exact test; for the disjoint Option-Dependent and Option-Independent subsets we use Fisher’s exact test. The paired comparisons between the 10-option baseline and each model’s worst $N=1000$ cell are reported in Table 9 and are significant for every model ($p < 10^{-4}$). Block also differs significantly from Sparse for every model ($p < 10^{-4}$).

At $N=1000$, Sparse, $w=0.75$, the dependency gap is also significant for every model (Fisher $p < 1.1 \times 10^{-6}$): GPT-OSS-120B 17.2 vs. 36.5; GPT-OSS-20B 4.6 vs. 13.4; Qwen3.5-27B 43.5 vs. 81.0; Qwen3.5-9B 24.3 vs. 61.3; Gemma-4-31B 43.1 vs. 77.3; and Gemini-3-Flash 58.5 vs. 86.1 (Option-Dependent vs. Option-Independent accuracy).

To address question difficulty in the reasoning-length analysis, we fit a linear probability model of correctness on within-model-standardized completion tokens. Question fixed effects are absorbed by demeaning, model-by-condition dummies control the 180 experimental cells, and standard errors are clustered by question. The completion-length coefficient is -0.0411 (cluster-robust SE 0.0025, 95% CI $[-0.046, -0.036]$, $p = 4.3 \times 10^{-60}$).

Model	N	$\bar{\sigma}$	$\sigma_{\max}$
GPT-OSS-120B	100	0.6	0.8
	500 [†]	1.0	1.3
	1000 [†]	1.7	2.4
Gemma-4-31B	100	0.4	0.5
	500 [†]	0.5	0.6
	1000 [†]	0.6	0.7
Qwen3.5-27B	100	0.5	0.6
	500 [†]	0.5	0.6
	1000 [†]	0.7	0.8

Table 10: **Seed-robustness summary across $N \in \{100, 500, 1000\}$ options and three models (seeds 42, 123, 456).** $\bar{\sigma}$ and $\sigma_{\max}$: mean and maximum standard deviation across the 10 (window, mode) conditions, computed over the three seeds. All values are in percentage points (% acc.). [†]100-question subset; $N=100$ uses the full 1,000-question benchmark.

G Robustness to Distractor Sampling

The main experiments use a single distractor-sampling seed (42) for all conditions. To verify that the reported positional-bias effects are not artefacts of this choice, we repeat all 10 (window, mode) configurations with two additional seeds (123 and 456) across all three models and pool sizes. The seed controls which $N-10$ distractors are drawn from the per-domain pool of candidates; gold-answer placement and the original 10 options are unaffected. Because constructing the augmented benchmark at each pool size requires an LLM-based distractor quality-control pipeline, exhaustive multi-seed runs are costly; we therefore evaluate the full 1,000-question benchmark at $N=100$ and a 100-question subset at $N \in \{500, 1000\}$.

Table 10 reports the mean and maximum seed-to-seed standard deviation across the 10 (window, mode) conditions. Across all 9 model- N combinations, $\sigma_{\max}$ never exceeds 2.4 pp (GPT-OSS-120B, $N=1,000$), while the positional bias in that same setting spans 60.5 pp. The results reported in the main text are therefore invariant to distractor sampling at every evaluated scale.

H Block-Mode Wrong-Pick Breakdown

In Block placement mode the nine wrong original options are kept contiguous with the gold, so a wrong pick can land in one of three regions: BEFORE (a distractor preceding the entire block), INSIDE (a wrong original within the block), or AFTER (a distractor following the block). Unlike sparse

mode—where every wrong original is isolated and the before/after split cleanly reflects positional preference—Block mode conflates option-type attraction (original vs. distractor) with positional bias, which is why RQ2 restricts to Sparse mode. Figure 12 shows the three-way breakdown across windows and list lengths. INSIDE dominates across all conditions: models preferentially select from within the original-option block rather than from distractors. As the block shifts to later positions (higher window), more distractors accumulate before the block and BEFORE rises substantially—an effect that amplifies with N . AFTER remains negligible throughout, consistent with the primacy pull observed in sparse mode.

I Per-Model Reasoning Effort Across Conditions

Section 6.4 establishes the aggregate picture: token counts grow with N and window position, and longer traces predict errors. Figure 13 zooms in on two dimensions not resolved in the main analysis: placement mode (Block vs. Sparse) and option-dependency label (Independent vs. Dependent).

Sparse Placement Consistently Inflates Reasoning Budgets

Across all six models, Sparse mode elicits more completion tokens than Block mode at $N=1000$, with per-model excess ranging from +4% (GPT-OSS-120B) to +21% (Gemma-4-31B). Heavy reasoners are the most affected in absolute terms: Qwen3.5-27B, Qwen3.5-9B, and Gemini-3-Flash each spend roughly 1,000 additional tokens under sparse placement ($\approx +10-15\%$). This is visible as a consistent upward shift of the Sparse curves relative to Block in every panel of Figure 13. The most natural explanation is structural: sparse augmentation distributes the original options uniformly throughout the list, preventing the model from exploiting local contiguity to narrow its search early.

Option Dependency Has Model-Specific and Opposing Effects

The dependent/independent split yields no single direction. Models in the heavy-reasoner family diverge: Gemini-3-Flash and Qwen3.5-9B spend more tokens on dependent questions (+15% and +6% respectively), while Qwen3.5-27B shows virtually no difference (−4%). The two GPT-OSS models show the opposite pattern: 120B and 20B use fewer tokens on dependent questions (−23% and −20%), suggesting that they short-circuit search after detecting an

Name	Checkpoint	Access	Link
Qwen3.5-27B	Qwen/Qwen3.5-27B-FP8	Local (vLLM)	HuggingFace
Qwen3.5-9B	Qwen/Qwen3.5-9B	Together AI API	Together AI
Gemini-3-Flash	gemini-3-flash-preview	Google Vertex AI API	Google AI
Gemma-4-31B	google/gemma-4-31b-it	Together AI API	Together AI
GPT-OSS-120B	openai/gpt-oss-120b	Together AI API	Together AI
GPT-OSS-20B	openai/gpt-oss-20b	Together AI API	Together AI

Table 11: Models used in our experiments.

option-dependent cue rather than deliberating further. Gemma-4-31B is the most sensitive overall, investing +32% more tokens on dependent questions. The absence of a uniform direction implies that option dependency modulates reasoning effort through model-specific strategies rather than a shared mechanism.

J Further Strategy Resolution Details

J.1 Taxonomy Derivation and Classification Protocol

The S1–S10 taxonomy was induced from manual review rather than specified from outcomes: we examined traces stratified by model, N , and window, iteratively merged overlapping codes, and froze the definitions in Table 19 before automated labeling. This interpretability-oriented perspective is conceptually related to descriptive text-mining approaches that recover recurring, statistically supported phenomena from unlabeled text collections (Frisoni and Moro, 2020).

To select the classifier used for large-scale annotation, we piloted three candidate models—GPT-OSS-20B, GPT-OSS-120B, and Qwen3.5-9B—on a held-out subset of the manually coded traces. Qwen3.5-9B achieved the highest agreement with the human annotations (Cohen’s $\kappa = 0.86$), compared with 0.79 for GPT-OSS-120B and 0.71 for GPT-OSS-20B, and was therefore selected to classify the full stratified sample. The reported audit covers 100 questions crossed with the 30 augmented configurations for each model (3,000 traces per model; 18,000 total). The classifier receives the question, candidate configuration, full reasoning trace, and the fixed S1–S10 definitions, and must return exactly one strategy code; plots in the main text merge S5–S10 into *Other*.

While Section 6.3 focuses on sparse placement averaged over window positions, we provide here three complementary views. Figure 11 reports the strategy distribution for Block placement averaged over window positions, separated by de-

pendency class. Figures 9 and 10 show the full window-position breakdown for sparse and block mode respectively, at each option count $N \in \{100, 500, 1000\}$.

Setting	Overall	Domains				Dependency		Error Type	
		CS	Health	Law	Math	Opt-Dep	Opt-Ind	Original	Random
10 options									
Baseline	83.0	90.0	77.2	70.8	94.0	72.8	91.7	-	-
Block									
100 options									
0.00	81.7	91.2	73.2	68.4	94.0	69.8	91.8	99.3	0.7
0.25	81.2	89.6	71.2	70.8	93.2	70.4	90.4	98.5	1.5
0.50	80.4	89.2	72.0	68.0	92.4	68.7	90.4	97.8	2.2
0.75	81.2	89.6	74.0	69.2	92.0	69.1	91.5	98.0	2.0
1.00	80.9	90.0	72.4	68.0	93.2	70.0	90.2	98.3	1.7
500 options									
0.00	81.2	89.6	72.0	69.6	93.6	69.8	90.9	98.6	1.4
0.25	79.6	88.4	70.4	67.2	92.4	67.0	90.4	96.3	3.7
0.50	79.3	88.4	69.6	66.0	93.2	66.5	90.2	96.0	4.0
0.75	79.9	90.0	71.6	65.6	92.4	67.4	90.6	96.3	3.7
1.00	80.1	89.6	70.8	66.4	93.6	68.0	90.4	95.9	4.1
1000 options									
0.00	80.5	88.8	73.2	68.8	91.2	68.7	90.6	99.3	0.7
0.25	78.8	88.4	70.0	65.2	91.6	66.1	89.6	95.2	4.8
0.50	78.7	87.2	69.6	67.6	90.4	67.4	88.3	94.3	5.7
0.75	78.2	87.6	68.8	65.2	91.2	65.2	89.3	94.8	5.2
1.00	79.6	89.2	72.8	65.2	91.2	68.3	89.3	95.4	4.6
Sparse									
100 options									
0.00	82.2	92.0	72.8	70.8	93.2	71.5	91.3	98.7	1.3
0.25	81.7	90.0	70.4	73.6	92.8	69.6	92.0	98.8	1.2
0.50	80.5	91.2	70.8	66.0	94.0	69.6	89.8	98.8	1.2
0.75	78.5	88.0	68.4	65.6	92.0	65.2	89.8	98.5	1.5
1.00	78.0	89.2	66.8	64.4	91.6	63.9	90.0	98.5	1.5
500 options									
0.00	84.3	91.6	77.6	74.8	93.2	75.0	92.2	97.4	2.6
0.25	76.9	87.2	65.2	64.0	91.2	63.9	88.0	95.2	4.8
0.50	76.1	84.0	67.2	60.8	92.4	62.0	88.1	95.1	4.9
0.75	74.6	86.4	62.4	59.4	90.0	59.7	87.2	95.1	4.9
1.00	79.1	87.6	66.4	70.8	91.6	66.3	90.0	96.0	4.0
1000 options									
0.00	86.0	93.2	78.8	76.4	95.6	77.0	93.7	96.0	4.0
0.25	76.5	85.6	65.9	63.6	90.8	63.0	88.0	93.9	6.1
0.50	73.3	82.4	60.4	60.8	89.6	59.4	85.2	92.7	7.3
0.75	73.4	81.6	60.4	62.0	89.6	58.5	86.1	93.4	6.6
1.00	76.0	84.0	67.6	62.3	90.0	63.4	86.7	93.0	7.0

Table 12: **Accuracy of Gemini-3-Flash across all experimental conditions.** Rows are grouped by placement mode (Block / Sparse) and option count $N \in \{100, 500, 1000\}$; the window position (0.00–1.00) indicates where the gold option is placed within the option list. Columns report overall accuracy, four domain slices, the option-dependency split, and the error-type breakdown (errors on original wrong options vs. random distractors). The 10-option baseline uses the original MMLU-Pro options without augmentation.

Setting	Overall	Domains				Dependency		Error Type	
		CS	Health	Law	Math	Opt-Dep	Opt-Ind	Original	Random
10 options									
Baseline	77.7	88.8	71.0	59.5	91.9	64.2	89.4	-	-
Block									
100 options									
0.00	78.0	89.6	68.3	61.6	93.1	64.2	89.8	99.6	0.4
0.25	76.0	90.0	61.5	59.6	93.8	61.2	88.8	98.9	1.1
0.50	75.6	87.6	66.0	56.8	92.7	61.9	87.5	98.8	1.2
0.75	75.1	88.6	64.0	56.2	91.9	59.3	88.6	98.0	2.0
1.00	76.2	89.2	67.1	56.4	92.7	61.2	89.1	98.9	1.1
500 options									
0.00	76.6	89.7	68.4	57.6	91.5	62.5	88.8	98.7	1.3
0.25	72.5	86.2	60.4	52.4	91.8	56.6	86.2	95.3	4.7
0.50	73.1	86.9	62.4	51.6	92.5	55.9	88.1	95.4	4.6
0.75	72.2	86.5	61.6	50.4	91.3	55.2	86.9	95.1	4.9
1.00	71.9	85.2	57.6	53.2	92.6	54.5	86.9	95.5	4.5
1000 options									
0.00	76.6	87.4	68.0	58.8	92.7	63.0	88.4	98.6	1.4
0.25	72.8	85.4	61.0	54.4	90.9	58.2	85.3	94.4	5.6
0.50	70.0	83.3	57.8	51.0	89.0	50.9	86.6	92.0	8.0
0.75	69.9	85.2	56.6	51.0	88.3	52.0	85.4	93.0	7.0
1.00	70.6	84.2	58.0	50.6	90.1	51.9	86.7	93.9	6.1
Sparse									
100 options									
0.00	80.5	90.6	70.0	68.8	92.8	67.8	91.4	98.8	1.2
0.25	76.3	89.6	65.7	57.4	93.1	60.6	89.8	99.2	0.8
0.50	72.5	85.9	57.3	57.2	90.2	56.7	86.2	98.9	1.1
0.75	71.7	86.8	55.2	54.0	91.5	54.3	86.7	99.1	0.9
1.00	71.4	88.2	55.2	51.8	91.0	53.9	86.4	98.5	1.5
500 options									
0.00	84.6	92.3	77.6	72.8	95.9	74.5	93.3	96.8	3.2
0.25	72.0	87.6	57.6	52.8	91.2	54.7	87.0	96.0	4.0
0.50	67.4	86.4	54.0	41.6	88.8	47.4	84.8	95.6	4.4
0.75	64.3	80.2	48.4	40.8	88.8	43.0	82.6	94.3	5.7
1.00	66.8	84.2	52.2	41.6	91.2	47.4	83.8	95.8	4.2
1000 options									
0.00	87.7	93.1	81.1	80.7	95.9	80.3	94.0	96.7	3.3
0.25	70.9	86.1	55.2	52.4	91.2	52.6	86.6	93.8	6.2
0.50	64.8	81.0	48.6	41.6	90.1	45.8	81.4	91.5	8.5
0.75	63.3	80.2	46.2	41.6	88.0	43.5	81.0	92.0	8.0
1.00	66.4	78.0	48.2	49.6	91.1	49.8	80.8	92.5	7.5

Table 13: **Accuracy of Qwen3.5-27B across all experimental conditions.** Rows are grouped by placement mode (Block / Sparse) and option count $N \in \{100, 500, 1000\}$; the window position (0.00–1.00) indicates where the gold option is placed within the option list. Columns report overall accuracy, four domain slices, the option-dependency split, and the error-type breakdown (errors on original wrong options vs. random distractors). The 10-option baseline uses the original MMLU-Pro options without augmentation.

Setting	Overall	Domains				Dependency		Error Type	
		CS	Health	Law	Math	Opt-Dep	Opt-Ind	Original	Random
10 options									
Baseline	71.8	86.4	65.6	46.0	89.2	53.0	87.8	-	-
Block									
100 options									
0.00	71.1	85.2	64.0	44.8	90.4	52.4	87.0	99.4	0.6
0.25	67.0	81.0	57.9	40.7	88.6	48.4	83.1	97.0	3.0
0.50	66.4	80.7	56.5	40.8	87.9	48.1	81.9	96.3	3.7
0.75	63.8	77.1	56.0	39.1	83.1	46.4	78.6	94.3	5.7
1.00	64.0	76.2	54.8	39.0	85.9	45.9	79.5	94.1	5.9
500 options									
0.00	69.6	82.3	62.6	42.5	91.0	50.9	85.5	99.2	0.8
0.25	60.4	72.1	53.8	37.6	78.1	43.0	75.2	88.8	11.2
0.50	57.8	67.9	49.6	42.4	71.6	43.6	70.1	83.7	16.3
0.75	52.8	65.5	49.6	32.6	63.3	39.4	64.3	78.8	21.2
1.00	51.2	60.5	46.8	33.5	64.2	36.3	63.9	77.4	22.6
1000 options									
0.00	70.6	84.4	63.2	45.2	89.6	53.5	85.2	99.4	0.6
0.25	52.6	66.0	45.2	33.6	65.6	37.6	65.4	80.6	19.4
0.50	46.5	58.0	44.8	26.8	56.4	32.6	58.3	71.3	28.7
0.75	38.1	47.4	36.5	24.8	43.8	28.3	46.5	62.0	38.0
1.00	41.0	52.4	38.4	28.8	44.3	30.5	49.9	64.7	35.3
Sparse									
100 options									
0.00	77.6	87.8	73.3	58.9	90.5	64.2	89.1	98.1	1.9
0.25	68.1	81.1	58.1	44.0	89.4	51.3	82.4	96.9	3.1
0.50	61.5	76.3	49.8	34.5	85.1	42.2	77.7	97.0	3.0
0.75	58.9	74.0	47.2	31.3	83.7	38.3	76.5	96.6	3.4
1.00	55.0	71.2	44.0	25.6	79.4	35.5	71.6	96.0	4.0
500 options									
0.00	80.5	88.5	76.8	63.4	93.1	67.8	91.3	94.0	6.0
0.25	57.8	72.5	49.6	33.5	75.8	39.3	73.5	87.0	13.0
0.50	43.6	58.3	40.5	19.4	56.3	28.4	56.6	82.8	17.2
0.75	39.4	50.6	34.3	16.0	56.5	24.1	52.4	80.5	19.6
1.00	34.3	43.9	30.6	10.8	52.2	20.3	46.4	80.2	19.8
1000 options									
0.00	82.9	91.6	79.2	69.2	91.6	72.0	92.2	94.3	5.7
0.25	48.8	61.6	42.4	34.0	57.2	33.7	61.7	76.8	23.2
0.50	36.8	48.8	34.4	17.6	46.4	22.4	49.1	72.2	27.8
0.75	27.6	38.4	23.2	12.8	36.1	17.2	36.5	70.1	29.9
1.00	22.4	33.1	20.8	7.6	28.4	11.3	32.0	66.9	33.1

Table 14: **Accuracy of GPT-OSS-120B across all experimental conditions.** Rows are grouped by placement mode (Block / Sparse) and option count $N \in \{100, 500, 1000\}$; the window position (0.00–1.00) indicates where the gold option is placed within the option list. Columns report overall accuracy, four domain slices, the option-dependency split, and the error-type breakdown (errors on original wrong options vs. random distractors). The 10-option baseline uses the original MMLU-Pro options without augmentation.

Setting	Overall	Domains				Dependency		Error Type	
		CS	Health	Law	Math	Opt-Dep	Opt-Ind	Original	Random
10 options									
Baseline	66.8	82.0	60.0	34.0	91.2	48.3	82.6	-	-
Block									
100 options									
0.00	66.2	80.4	60.0	34.0	90.4	48.0	81.7	99.5	0.5
0.25	64.6	80.0	56.8	35.2	86.4	46.3	80.2	96.5	3.5
0.50	62.2	75.2	56.0	32.8	84.8	47.2	75.0	94.4	5.6
0.75	58.4	70.8	53.2	29.6	79.9	41.5	72.7	93.1	6.9
1.00	58.2	72.0	50.8	30.5	79.2	41.6	72.2	91.3	8.7
500 options									
0.00	64.3	79.1	57.6	30.0	90.4	45.4	80.3	98.4	1.6
0.25	53.2	65.6	52.0	25.6	69.8	40.7	63.9	85.7	14.3
0.50	45.1	58.8	43.6	21.8	56.2	32.8	55.7	75.4	24.6
0.75	42.6	54.0	41.0	26.6	48.8	31.3	52.2	71.1	28.9
1.00	41.3	52.6	41.2	23.0	48.2	31.4	49.6	71.1	28.9
1000 options									
0.00	65.6	79.6	62.4	30.4	90.0	47.6	80.9	99.1	0.9
0.25	36.0	46.0	38.4	20.8	38.8	27.4	43.3	64.1	35.9
0.50	29.0	39.2	28.8	17.6	30.5	20.6	36.2	57.7	42.3
0.75	23.3	36.1	28.4	11.6	17.2	15.4	30.1	47.0	53.0
1.00	31.3	38.0	37.2	17.2	32.9	21.7	39.5	57.2	42.8
Sparse									
100 options									
0.00	74.2	86.8	71.2	48.0	91.1	59.1	87.2	95.9	4.1
0.25	63.7	80.4	52.8	34.0	87.5	44.8	79.8	96.0	4.0
0.50	54.4	66.4	44.4	24.8	81.9	36.7	69.4	95.1	4.9
0.75	51.9	67.2	42.4	21.6	76.4	34.4	66.8	94.1	5.9
1.00	44.4	57.6	35.2	15.6	69.2	25.9	60.2	92.1	7.9
500 options									
0.00	76.5	88.3	73.5	53.0	91.2	61.4	89.4	92.0	8.0
0.25	44.8	54.8	45.2	18.8	60.6	32.7	55.2	80.1	19.9
0.50	30.6	45.8	26.4	12.8	37.4	20.7	39.0	74.8	25.2
0.75	27.8	39.2	27.2	9.2	35.6	16.7	37.2	73.4	26.6
1.00	25.0	34.9	22.8	12.8	29.5	16.9	32.0	71.9	28.1
1000 options									
0.00	79.7	89.2	76.0	62.8	90.8	67.6	90.0	88.4	11.6
0.25	22.7	28.4	24.0	9.6	28.8	14.6	29.6	56.6	43.4
0.50	16.2	18.1	22.0	8.0	16.8	10.0	21.5	52.1	47.9
0.75	9.3	13.6	11.2	2.4	10.0	4.6	13.4	50.4	49.6
1.00	13.3	12.8	21.6	4.0	14.9	7.6	18.2	54.1	45.9

Table 15: **Accuracy of GPT-OSS-20B across all experimental conditions.** Rows are grouped by placement mode (Block / Sparse) and option count $N \in \{100, 500, 1000\}$; the window position (0.00–1.00) indicates where the gold option is placed within the option list. Columns report overall accuracy, four domain slices, the option-dependency split, and the error-type breakdown (errors on original wrong options vs. random distractors). The 10-option baseline uses the original MMLU-Pro options without augmentation.

Setting	Overall	Domains				Dependency		Error Type	
		CS	Health	Law	Math	Opt-Dep	Opt-Ind	Original	Random
10 options									
Baseline	73.0	82.1	63.0	54.8	92.4	58.0	86.0	-	-
Block									
100 options									
0.00	71.1	82.7	60.8	48.0	92.8	53.7	85.9	98.9	1.1
0.25	67.9	83.1	54.8	46.0	88.0	50.9	82.5	97.3	2.7
0.50	65.7	79.8	54.0	41.2	88.0	46.7	82.0	97.1	2.9
0.75	66.3	79.0	54.8	43.6	88.0	49.8	80.5	95.9	4.1
1.00	67.1	81.9	52.4	47.6	86.8	49.6	82.2	96.0	4.0
500 options									
0.00	68.7	81.0	58.6	43.6	91.6	52.2	82.7	98.1	1.9
0.25	64.9	77.8	52.2	43.6	85.9	47.7	79.5	92.5	7.5
0.50	61.4	76.1	48.0	36.5	84.8	42.3	77.6	89.6	10.4
0.75	59.6	73.1	47.4	38.1	79.8	42.1	74.5	87.2	12.8
1.00	61.3	75.0	48.2	37.0	85.4	42.2	77.9	89.1	10.9
1000 options									
0.00	69.4	82.3	59.2	46.0	90.4	51.1	85.1	98.2	1.8
0.25	63.7	78.6	52.4	38.3	85.7	44.1	80.5	91.3	8.7
0.50	57.6	69.8	45.2	36.8	79.4	39.9	72.7	85.0	15.0
0.75	54.9	71.4	42.2	27.5	79.8	35.2	72.1	81.1	18.9
1.00	58.2	73.5	44.6	32.8	82.9	39.0	74.9	85.6	14.4
Sparse									
100 options									
0.00	74.4	82.3	67.6	55.6	92.0	60.4	86.2	97.7	2.3
0.25	69.5	83.5	56.2	49.2	89.2	52.2	84.4	97.4	2.6
0.50	63.2	76.5	47.6	40.8	88.0	45.0	78.8	97.3	2.7
0.75	60.3	76.3	44.0	35.2	85.6	40.9	76.8	97.5	2.5
1.00	61.7	78.3	46.0	35.2	87.5	41.5	79.0	97.1	2.9
500 options									
0.00	78.5	89.9	73.5	58.4	92.4	65.4	89.8	95.0	5.0
0.25	59.9	75.3	48.0	33.7	82.7	39.7	77.0	91.3	8.7
0.50	56.0	72.7	40.0	28.0	83.5	33.7	75.1	90.4	9.6
0.75	51.0	64.9	38.4	22.0	79.0	31.3	67.9	89.0	11.0
1.00	54.7	73.1	38.1	25.8	81.8	32.5	73.8	90.4	9.6
1000 options									
0.00	82.0	90.3	78.3	66.7	92.7	71.8	90.6	94.1	5.9
0.25	57.5	67.1	45.8	38.3	80.0	41.0	71.8	86.2	13.8
0.50	50.9	67.5	34.0	24.0	79.5	29.1	69.6	86.0	13.9
0.75	44.2	57.5	30.4	16.5	73.6	24.3	61.3	82.5	17.5
1.00	51.6	66.7	37.4	22.9	80.7	33.1	67.5	86.4	13.6

Table 16: **Accuracy of Qwen3.5-9B across all experimental conditions.** Rows are grouped by placement mode (Block / Sparse) and option count $N \in \{100, 500, 1000\}$; the window position (0.00–1.00) indicates where the gold option is placed within the option list. Columns report overall accuracy, four domain slices, the option-dependency split, and the error-type breakdown (errors on original wrong options vs. random distractors). The 10-option baseline uses the original MMLU-Pro options without augmentation.

Setting	Overall	Domains				Dependency		Error Type	
		CS	Health	Law	Math	Opt-Dep	Opt-Ind	Original	Random
10 options									
Baseline	79.3	89.2	68.8	67.3	92.0	64.5	91.8	-	-
Block									
100 options									
0.00	78.9	88.0	69.6	67.2	90.8	65.2	90.6	99.1	0.9
0.25	77.3	87.2	63.6	66.0	92.4	62.6	89.8	98.8	1.2
0.50	77.2	87.9	63.2	66.7	91.2	62.2	90.0	97.9	2.1
0.75	76.9	89.1	64.7	62.4	91.6	61.4	90.1	98.3	1.7
1.00	76.6	88.8	64.4	61.2	92.0	60.4	90.4	98.4	1.6
500 options									
0.00	77.5	88.0	68.4	61.9	91.6	61.9	90.7	98.3	1.7
0.25	74.1	87.5	59.6	58.6	90.4	57.5	88.1	95.3	4.7
0.50	74.2	86.7	62.8	57.6	90.0	58.3	87.9	93.9	6.1
0.75	72.5	83.9	58.1	60.2	87.5	57.0	85.6	93.6	6.4
1.00	74.6	85.6	61.5	64.0	87.2	60.1	86.9	95.6	4.4
1000 options									
0.00	78.9	89.2	68.4	66.8	91.2	64.1	91.5	98.6	1.4
0.25	72.1	85.8	59.2	56.0	87.0	55.9	85.9	93.6	6.4
0.50	72.1	84.2	58.7	59.2	86.2	56.2	85.5	93.3	6.7
0.75	70.3	83.6	59.0	55.1	83.2	55.1	83.2	92.0	8.0
1.00	73.4	87.8	61.1	57.8	86.7	58.1	86.4	94.3	5.7
Sparse									
100 options									
0.00	77.1	90.0	63.6	62.8	92.0	60.9	90.9	98.7	1.3
0.25	77.0	87.9	64.4	65.1	90.4	62.1	89.6	98.9	1.1
0.50	75.5	88.0	60.8	61.6	91.6	59.4	89.2	99.0	1.0
0.75	74.4	86.4	61.2	59.6	90.4	58.2	88.1	98.8	1.2
1.00	74.0	86.8	62.4	56.0	90.8	55.6	89.6	98.7	1.3
500 options									
0.00	82.3	92.3	74.5	68.6	93.5	69.6	93.1	97.2	2.9
0.25	73.4	86.8	58.6	58.1	90.0	55.3	88.7	95.7	4.3
0.50	65.7	81.6	48.8	47.4	84.7	45.5	82.7	93.5	6.5
0.75	65.5	81.5	50.0	46.6	83.6	47.4	80.8	93.4	6.6
1.00	69.2	82.3	54.0	56.9	83.5	53.7	82.3	93.8	6.2
1000 options									
0.00	83.9	94.3	74.6	72.2	93.9	72.8	93.2	96.4	3.6
0.25	66.2	77.0	54.4	49.6	83.5	49.8	79.7	90.5	9.5
0.50	66.2	79.0	53.5	49.4	82.5	49.0	80.8	90.6	9.4
0.75	61.7	75.9	47.0	42.0	81.3	43.1	77.3	90.5	9.5
1.00	65.8	80.6	50.4	49.6	82.3	49.0	79.8	90.6	9.4

Table 17: **Accuracy of Gemma-4-31B across all experimental conditions.** Rows are grouped by placement mode (Block / Sparse) and option count $N \in \{100, 500, 1000\}$; the window position (0.00–1.00) indicates where the gold option is placed within the option list. Columns report overall accuracy, four domain slices, the option-dependency split, and the error-type breakdown (errors on original wrong options vs. random distractors). The 10-option baseline uses the original MMLU-Pro options without augmentation.

Question	Gold	Distractor	Replacement	Reason
<i>Computer Science</i>				
Let a undirected graph G with edges $E = \langle 0,1 \rangle, \langle 4,1 \rangle, \langle 2,0 \rangle, \langle 2,1 \rangle, \langle 2,3 \rangle, \langle 1,3 \rangle$ which $\langle A,B \rangle$ represent Node A...	[1, 2]	2 1	3 4	The option "2 1" lists the vertices 1 and 2 (just in a different order), which is semantically equivalent to the correct minimum vertex cover [1, 2]. A domain...
Which is the smallest asymptotically?	$O(1)$	8.00	$O(n^2)$	A constant numeric value (8.00) represents $O(1)$ time, which is the same asymptotic class as the correct answer.
Can a stream cipher have perfect secrecy?	No, since the key is shorter than the message	It depends on the size of K	It always has perfect secrecy regardless of key size.	The answer "It depends on the size of K" could be interpreted as a correct statement about perfect secrecy, since perfect secrecy is achievable only when the...
<i>Mathematics</i>				
Which of the following is least likely to reduce bias in a sample survey?	Using a non-random sampling method	conducting the survey online targeting only the mall's registered customers	Using a stratified random sampling method	The distractor describes a specific non-random sampling method (selection bias), which is a correct answer to the question as it would not reduce bias.
If $x - 3 = 6$, what is the value of x?	9	9 kg/m	15 kg/m	The distractor contains the correct numerical value (9) followed by units. In many contexts, this could be considered numerically equivalent or partially...
A store has 3 boxes of shirts. Each box has 4 packages with 7 shirts in each package. The expression $3 \times (4 \times \dots)$	12×7	84	$3 + 4 + 7$	The distractor 84 is numerically equivalent to the correct answer ($12 \times 7 = 84$) and is the total number of shirts. Since the question asks for an expression...
<i>Law</i>				
A truck and a car collided at an intersection. Both drivers were injured, and each one filed a lawsuit...	Neither party recovers anything from the other.	recover nothing, because he was more at fault than the driver of the truck.	The truck driver must pay the car driver \$5,000 because he was twice as...	The option states that the plaintiff recovers nothing, which matches the correct outcome of no recovery, making it factually equivalent to the correct answer.
A man worked as a clerk in a gourmet coffee shop. His friend often stopped by the shop to chat with the man...	prevail, because the company would be vicariously liable for the man's...	prevail, because the employee was on duty at the time of the accident.	prevail, because the customer signed a waiver releasing the shop from any...	Option 0 states the customer will prevail due to the employee being on duty, which is essentially the correct legal reasoning (vicarious liability) and thus...
A plaintiff sued an insurance company to recover life insurance proceeds. At trial, the issue before the...	A sworn affidavit from the insured's brother verifying the insured's birth date.	sustain the objection, because the testimony is hearsay.	admit the testimony, because it is a present recollection recorded...	This option argues the evidence is inadmissible on hearsay grounds, which aligns with the correct answer that a sworn affidavit from a brother is inadmissible.
<i>Health</i>				
The mucosa of the upper larynx is innervated by the	internal laryngeal nerve which is the afferent limb of the cough reflex.	The vagus nerves	The phrenic nerves	The vagus nerves give rise to the internal laryngeal nerve, so a respondent could argue that "vagus nerves" is a correct (though imprecise) answer to the...
Which of the following structures accompany the oesophagus through the diaphragm?	The vagus nerves	Proprioceptive information is carried by the glossopharyngeal and vagus nerves.	The phrenic nerves accompany the oesophagus through the diaphragm.	The option mentions the vagus nerve, which is the correct structure that accompanies the oesophagus through the diaphragm, making it factually correct.
A 17-year-old girl is brought to the emergency department 15 minutes after being stung by a bee...	β 2-Adrenergic agonist	Adrenaline.	Ibuprofen.	Adrenaline (epinephrine) is a drug used in anaphylaxis and could be argued as an appropriate treatment for this...

Table 18: **Examples of flagged distractors and their corresponding replacements across different domains.** The *Reason* column summarizes the logical overlap identified by the LLM judge during the screening process.

Code	Name	Description
S1	Independent Derivation then Scan	The model solves the problem analytically (math, law, or logic) in its reasoning trace before looking at the options. Once a result is derived, it explicitly searches the long list of candidates to locate and select the matching entry.
S2	Sequential Order Review	The model methodically evaluates options one by one, usually starting from the first index (0) or occasionally from the end (999). It judges each statement's validity against its internal knowledge until it identifies a match or narrows the field.
S3	Eliminative Filtering	The model scans the option list to discard obviously incorrect, irrelevant, or nonsensical candidates. This narrows the search space to a manageable subset (shortlist) before the final selection is made.
S4	Candidate-Led Shortlisting	The model brainstorms a set of plausible internal answers (candidates) based on the question before scanning the list. It then specifically searches the long option list to see which of its internally generated candidates actually appear.
S5	Surface Lexical Matching	The model identifies specific key phrases or unique identifiers in the prompt and performs a heuristic search for those same words in the option list. This relies on textual similarity rather than full analytical derivation.
S6	Direct Outcome Recognition	The model derives the answer or recalls the fact and identifies the correct option via direct mapping or recognition without verbalizing a search or scan of other indices.
S7	Index / Anchor Heuristic	The model selects an option based on its numerical index or list position rather than on semantic evaluation. The choice is driven by round-number attraction (e.g. 0, 500, 999) or a prior belief that answers cluster at certain positions, with no analytical justification given.
S8	Block / Cluster Detection	The model notices that a subset of options forms a semantically coherent group or a contiguous block and focuses evaluation on that cluster rather than the full list. Expected to be more common in block placement mode where the original options are contiguous.
S9	Explicit Overload Declaration	The model explicitly acknowledges that the number of options exceeds what it can evaluate in detail and announces a fallback plan. This is a meta-statement about processing limits that precedes another strategy (typically S7 or S10).
S10	Capitulation / Default Guess	The model produces a minimal or arbitrary answer with no genuine evaluation — picking index 0, a round number, or the first plausible option encountered. Typically follows an overload declaration or appears when the model fails to derive an answer and resorts to a token response.

Table 19: **Taxonomy of reasoning strategies used by LLMs on large-scale multiple-choice questions.** Strategies are ordered by overall frequency (S1 = most frequent).

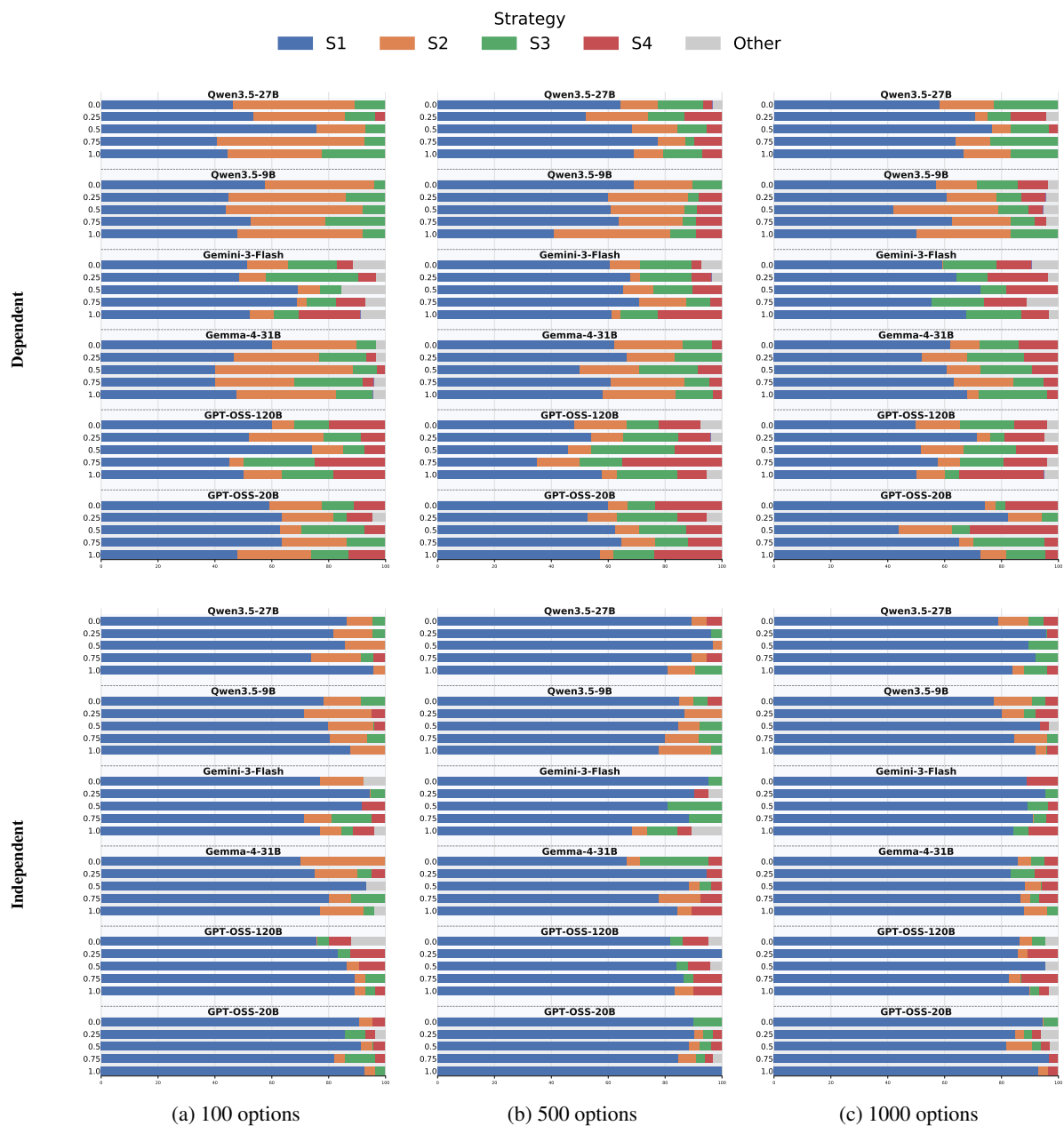


Figure 9: **Strategy variations across different candidate set sizes and window position for Sparse mode.** Comparison between dependent and independent settings with 100, 500, and 1000 candidate options.

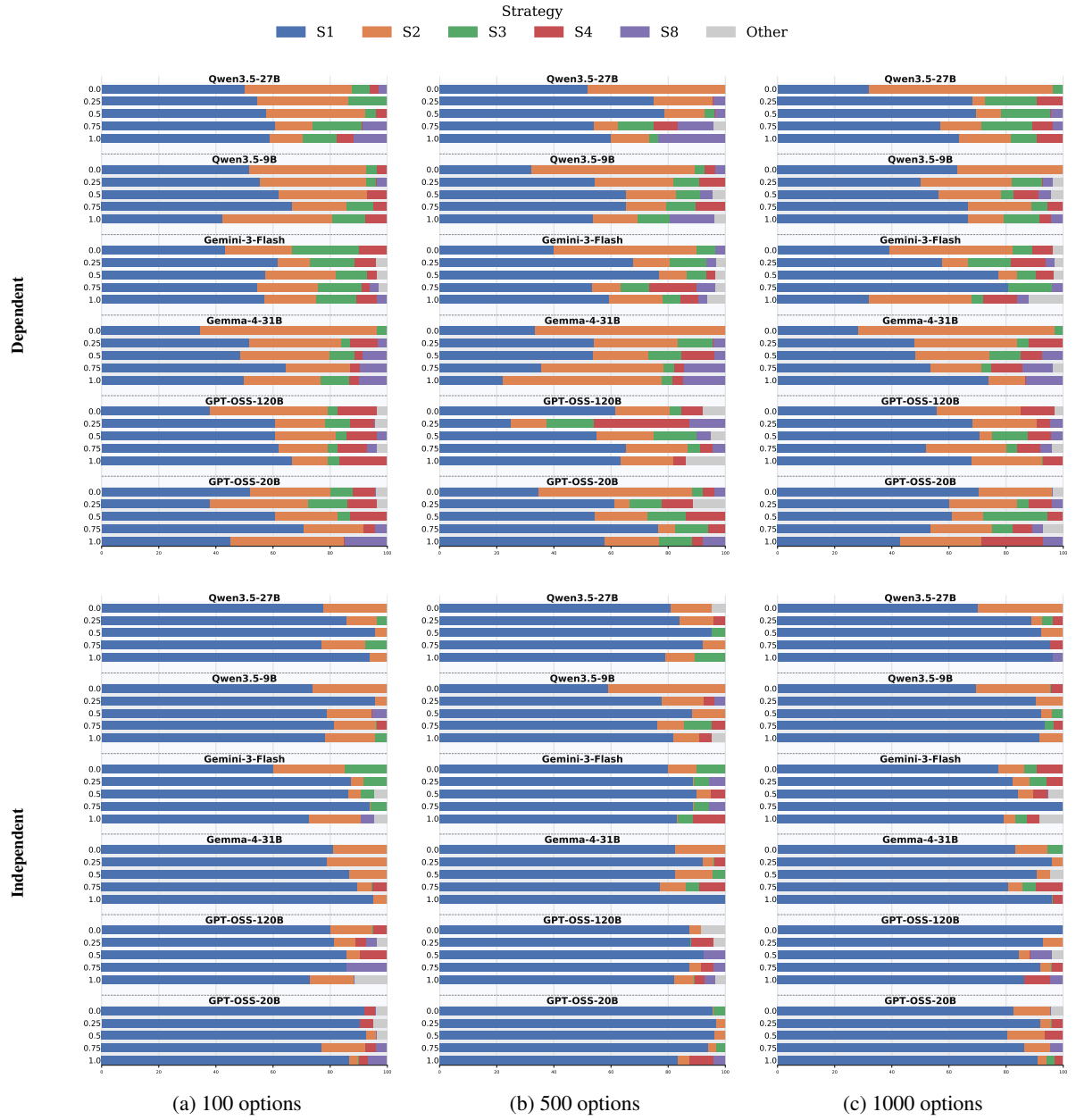
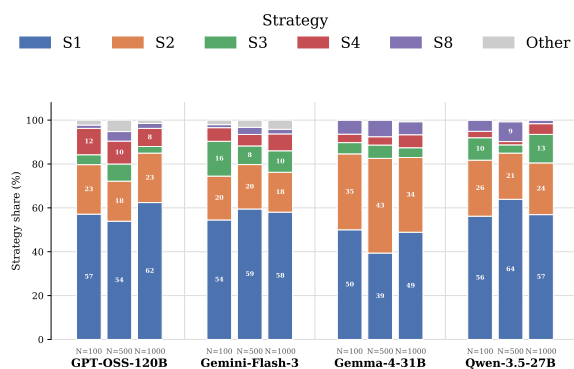
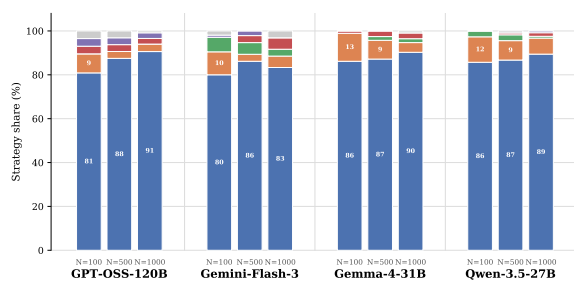


Figure 10: **Strategy variations across different candidate set sizes and window position for Block mode.** Comparison between dependent and independent settings with 100, 500, and 1000 candidate options.



(a) Option Dependent.



(b) Option Independent.

Figure 11: **Strategy distribution by question type and option count.** Each cluster shows three stacked bars ($N=100, 500, 1000$), one per model; Block placement, averaged over window positions.

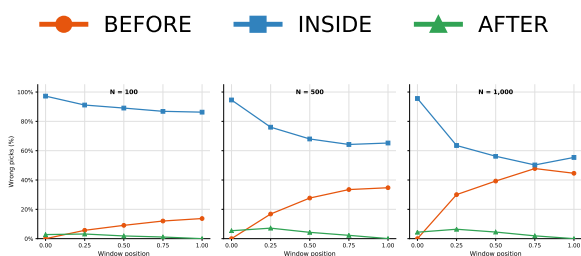


Figure 12: **Block-mode wrong-pick breakdown by region.** Before: orange circles; Inside: blue squares; After: green triangles.

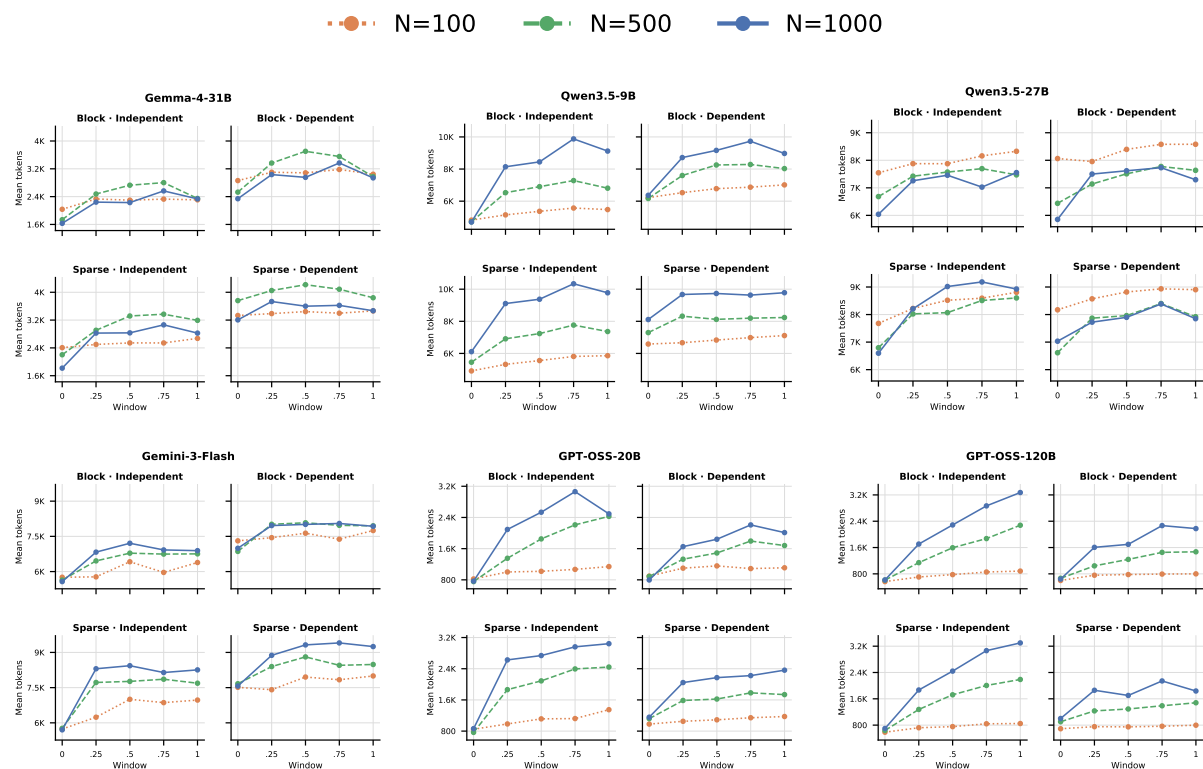


Figure 13: Mean reasoning tokens per window position for each model, broken down by placement mode (Block vs. Sparse, rows) and option-dependency label (Independent vs. Dependent, columns). Lines within each panel correspond to different distractors counts ($N \in \{100, 500, 1000\}$).

Prompt Template — Question Dependency Classification (All Three Judges)

Decide whether a knowledgeable person could answer the question correctly **WITHOUT** seeing the answer options.

Core test — mentally remove the options and ask the question as open-ended:

- If answerable (a specific fact, definition, formula, calculation, etc.) → option_independent
- If the answer is ambiguous or requires comparing the options → option_dependent

Additional rules:

- (1) Phrasing like “which of the following” does NOT automatically make a question option-dependent.
- (2) EXCEPT / NOT / LEAST / best-among framings are almost always option-dependent.
- (3) Options introducing evaluation categories absent from the stem (e.g. “Not Given”, “All of the above”) make the question option-dependent.
- (4) When in doubt, prefer the label for which domain knowledge alone suffices.

[5 few-shot examples omitted for brevity]

Question:

<question text>

Options:

<option list>

Return ONLY valid JSON, no explanation, no markdown:

{“label”: “option_dependent” or “option_independent”}

Figure 14: Classification prompt submitted to all three judges (GPT-OSS-120B, Gemma-4-31B, Qwen3.5-9B). The full prompt includes five few-shot examples covering the main edge cases: open-ended phrasing, EXCEPT/NOT framing, best-among comparisons, and composite True/False/Not-Given option sets.

Prompt Template — Distractor Safety Sweep (GPT-OSS-120B / Gemini-3-Flash)

You are evaluating answer options for a multiple-choice question.

Question: <question text>

Correct answer (for reference only): <gold option>

Candidate options (labelled 0-<N-1>): <distractor batch>

Task: For each option decide whether it could be considered a correct or defensibly correct answer.

An option is **UNSAFE** if:

- It is factually true as an answer to this question, even if phrased differently.
- A domain expert could reasonably argue it is correct or partially correct.
- It is semantically or numerically equivalent to the correct answer.

An option is **SAFE** if a domain expert would clearly reject it as wrong.

For every UNSAFE option also write a **replacement**: a new answer that is clearly wrong but plausible to someone without deep domain knowledge, in the same style and approximate length as the other options.

Respond with a JSON object only, no other text:

```
{
  "flagged": [
    {
      "id": <0-based index within this batch>,
      "reason": "<one or two sentences>",
      "replacement": "<new SAFE option text>"
    }
  ]
}
```

Return {"flagged": []} if all options are safe.

Figure 15: **Distractor safety prompt submitted to GPT-OSS-120B (CS, Law, Health) and Gemini-3-Flash (Mathematics).** Up to 100 distractors are submitted per call; the gold option is provided as a reference anchor so the judge can detect semantic equivalence.