LongBench-Pro: A More Realistic and Comprehensive Bilingual Long-Context Evaluation Benchmark

Anonymous ACL submission

Abstract

The rapid expansion of context length in large language models (LLMs) has outpaced existing evaluation benchmarks, many of which trade off scalability and realism: synthetic tasks underrepresent real-world complexity, while fully manual annotation is costly at extreme lengths and diverse scenarios. We present **LongBench Pro**, a more realistic and comprehensive bilingual benchmark of 1,500 naturally occurring long-context samples in English and Chinese spanning 11 primary tasks and 25 secondary tasks, with input lengths from 8k to 256k tokens. LongBench Pro supports fine-grained analysis with task-specific metrics and a multi-dimensional taxonomy of *context requirement* (full vs. partial dependency), *length* (six levels), and *difficulty* (four levels calibrated by model performance). To balance quality with scalability, we propose a *Human-Model Collaborative Construction* pipeline: frontier LLMs draft challenging questions and reference answers, along with design rationales and solution processes, to reduce the cost of expert verification. Experts then rigorously validate correctness and refine problematic cases. Evaluating 46 widely used long-context LLMs on LongBench Pro yields three findings: (1) long-context optimization contributes more to long-context comprehension than parameter scaling; (2) effective context length is typically shorter than the claimed context length, with pronounced cross-lingual misalignment; and (3) the "thinking" paradigm helps primarily models trained with native reasoning, while mixed-thinking designs offer a promising Pareto trade-off. In summary, LongBench Pro provides a robust testbed for advancing long-context understanding.

1 Introduction

Understanding and reasoning over long contexts has become a core capability of large language models (LLMs). With advances in architectures and computational resources, context length has continuously expanded (Comanici et al., 2025; OpenAI, 2025; Anthropic, 2025), making it possible for LLMs to tackle complex long-context tasks such as large-scale codebase analysis and systematic literature review. This rapid progress demands that long-context benchmarks evolve swiftly to accurately assess models' true long-context capabilities and facilitate open-source models in closing the gap with closed-source models.

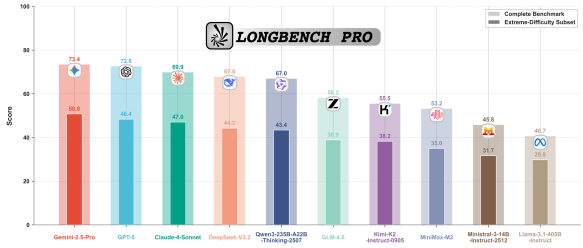


Figure 1: Results of modern LLMs on LongBench Pro.

Several long-context evaluation benchmarks exist. Synthetic benchmarks (Hsieh et al., 2024; Yen et al., 2024) enable controlled evaluation at scale, while manually annotated benchmarks (Qiu et al., 2024; Bai et al., 2025) ensure authenticity through complex tasks and realistic scenarios. Although these benchmarks have guided model development over the past few years, they are increasingly insufficient (in terms of task coverage, difficulty level, etc.) for evaluating next-generation models, as context lengths extend to millions of tokens and model capabilities continue to advance rapidly. This creates a pressing need for new benchmarks that satisfy more authentic and comprehensive evaluation of diverse long-context capabilities.

To address this urgent need, we introduce **LongBench Pro**, a realistic and comprehensive long-context evaluation benchmark containing 1,500 bilingual samples in English and Chinese. Built entirely on authentic, naturally occurring long documents, LongBench Pro encompasses 11 primary and 25 secondary tasks, covering the full spectrum of long-context capabilities assessed by existing

Benchmark	Text Type	#Task	Metric	Language	Dimensional Categorization		
					Ctx-Req	Length	Difficulty
RULER (Hsieh et al., 2024)	Fully Synthetic	4	Single	EN	○	●	○
∞BENCH (Zhang et al., 2024)	Synthetic, Natural	6	Diverse	EN, ZH	○	○	○
CLongEval (Qiu et al., 2024)	Synthetic, Natural	7	Diverse	ZH	●	◐	○
HELMET (Yen et al., 2024)	Synthetic, Natural	7	Diverse	EN	○	●	○
LongBench v2 (Bai et al., 2025)	Fully Natural	6	Single	EN	○	◐	◐
LongBench Pro (Ours)	**Fully Natural**	**11**	**Diverse**	**EN, ZH**	●	●	●

Table 1: **Comparison of long-context benchmarks.** Ctx-Req denotes the Context Requirement dimension. ○ indicates the absence of dimensional categorization; ● indicates the presence of detailed categorization in this dimension (Ctx-Req ≥ 2; Length > 3; Difficulty > 2); ◐ indicates the presence of categorization with a coarser granularity that does not meet the fine-grained criteria.

benchmarks. It employs diverse evaluation metrics for fine-grained measurement and introduces a multi-dimensional taxonomy: (1) *Context Requirement—Full* context (global integration) versus *Partial* context (localized retrieval); (2) *Length*—six levels uniformly distributed from 8k to 256k tokens to analyze scaling behavior; and (3) *Difficulty*—four levels ranging from Easy to Extreme, calibrated based on model performance. Table 1 summarizes the key differences between LongBench Pro and existing benchmarks.

Creating such a benchmark requires moving beyond traditional construction approaches. Synthetic methods, despite their scalability, cannot fully capture the semantic complexity of realistic long context scenarios. Conversely, purely manual annotation becomes prohibitively expensive and cognitively demanding at extreme lengths, limiting practicality and scalability. To achieve both authenticity and scalability, we propose a novel *Human-Model Collaborative Construction* strategy that synergizes the strengths of advanced LLMs and human expertise. In this strategy, models generate challenging questions and reference answers, along with design rationales and solution processes, from authentic, long documents corresponding to realistic, long-context scenarios. Meanwhile, human experts act as rigorous critics to verify correctness, filter flawed samples, and calibrate difficulty levels. This approach achieves high realism through the use of authentic documents, ensures quality through expert verification, and reduces costs through model involvement, enabling the efficient production of challenging, comprehensive, and high-quality evaluation samples.

We conduct extensive evaluations of 46 widely studied long-context LLMs on LongBench Pro and derive three main findings:

(1) Long-context optimization surpasses parameter scaling for improving long-context comprehension. Performance gains from extending effective context length significantly exceed those from scaling model size. This marks a shift from the traditional "scale-first" to a "context-optimization-first" paradigm.

(2) Context length–capability gaps and language bias constrain performance consistency. Models' effective context lengths often fall short of their claimed context lengths, and cross-lingual performance remains misaligned. While stronger models reduce these gaps, they persist across most long-context models.

(3) The thinking paradigm requires native training to overcome long-context bottlenecks. Thinking substantially improves long-context performance, but primarily in models trained with native reasoning. Conventional instruct models show limited or degraded gains when prompted to think. Mixed-thinking models combining fast response and deep reasoning achieve Pareto-optimal performance and may define the future paradigm.

Our contributions are threefold:

- We release **LongBench Pro**, a realistic and comprehensive bilingual benchmark with 1,500 samples and multi-dimensional categorization for rigorous long-context evaluation.
- We validate a **Human-Model Collaborative Construction** strategy that transcends the cost-quality trade-off, enabling high-quality sample generation at significantly lower cost than purely manual annotation.
- We provide **comprehensive analysis** of 46 long-context LLMs, revealing insights into context length scaling, cross-lingual performance, and reasoning mechanisms.

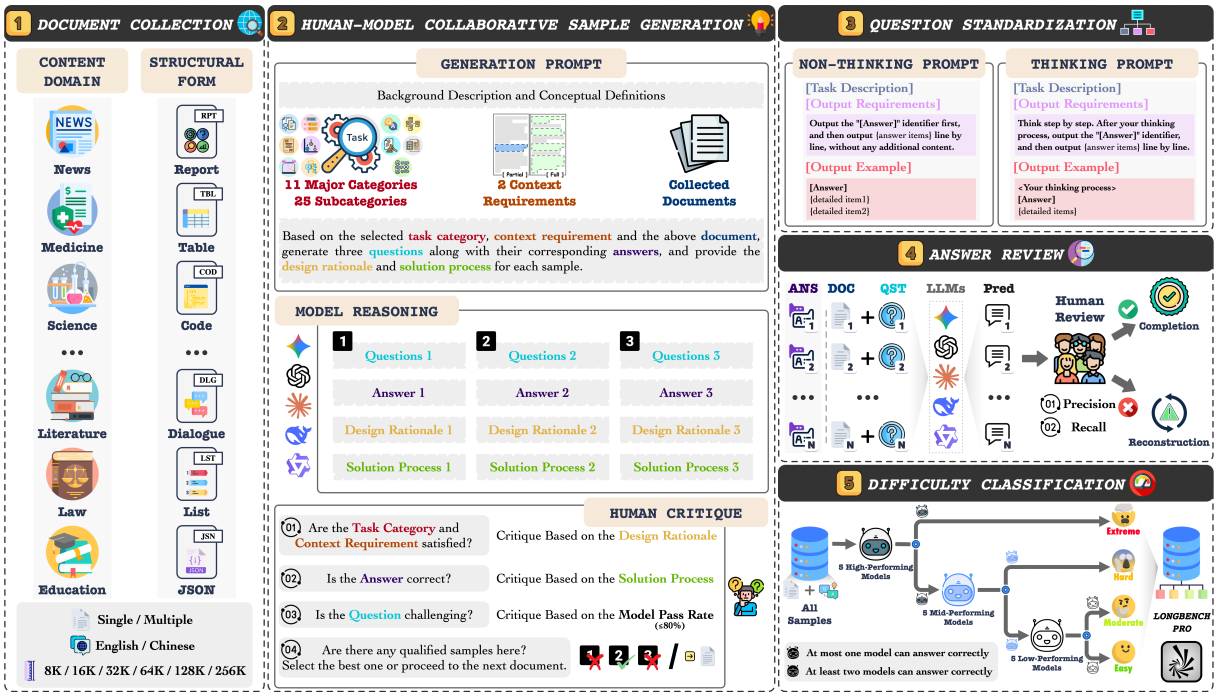


Figure 2: The construction process of LongBench Pro includes document collection, human–model collaborative sample generation, question standardization, answer review, and difficulty classification.

2 Task Framework of LongBench Pro

To systematically characterize long-context capabilities, we organize LongBench Pro into a two-level task taxonomy. We consolidate task formulations from prior long-context benchmarks into 11 primary categories. Appendix A maps our taxonomy to existing benchmark tasks, illustrating that LongBench Pro covers all the core capability dimensions evaluated by existing benchmarks.

Beyond task type, we introduce *context requirement*, an orthogonal dimension capturing how globally a model must depend on the input document:

- **Full**: solving the task requires integrating evidence dispersed across multiple, distant spans of the document, emphasizing integration and reasoning;
- **Partial**: solving the task primarily relies on localized spans, emphasizing localization and retrieval.

Crossing the 11 primary categories with context requirements to design 25 secondary categories. Table 2 lists the full taxonomy, and Appendix B provides detailed task definitions.

3 Construction Process of LongBench Pro

Note: The prompts and guidelines involved in the construction process are detailed in Appendix C.

3.1 Document Collection

To ensure realism and coverage, we curate naturally occurring long documents from the public internet across diverse domains (e.g., news, medicine, science, literature, law, and education) and formats (e.g., reports, tables, code, dialogues, lists, and JSON). We balance the collection across single-document and multi-document settings, as well as English and Chinese, and six target length buckets (8k/16k/32k/64k/128k/256k tokens), where token length is measured using the Qwen tokenizer (Yang et al., 2025). Since raw documents rarely match target lengths exactly, we assign a document to a bucket if its length falls within $\pm 20\%$ of the target. All documents undergo a compliance review by human annotators to exclude content that is privacy-sensitive, copyrighted, or otherwise non-compliant.

3.2 Human-Model Collaborative Sample Generation

To balance authenticity with cost, we adopt a human-model collaborative construction strategy. Given a document, we prompt multiple frontier LLMs (Gemini-2.5-Pro (Comanici et al., 2025), GPT-5 (OpenAI, 2025), Claude-4-Sonnet (Anthropic, 2025), DeepSeek-V3.2 (Liu et al., 2025), and Qwen3-235B-A22B-Thinking-2507 (Yang et al., 2025)) to draft three candidate samples

Task	Description	Context Requirement	Metric
T1 Retrieval & Ranking	Retrieve content and rank most relevant first		
T1.1 Global Cohesive Retrieval	Retrieve full text and reorganize	Full	NDCG@k
T1.2 Key-Snippet Retrieval	Locate target fragment in specified paragraph	Partial	NDCG@k
T2 Sequencing & Structure Reconstruction	Restore timeline or logical order		
T2.1 Global Timeline Reconstruction	Sort unordered events in the whole text	Full	Pairwise Accuracy
T2.2 Local Causal Chain Sorting	Sort content in a specific paragraph	Partial	Pairwise Accuracy
T3 Evidence-Grounded QA	Answer fact/reasoning questions based on evidence		
T3.1 Multi-Doc Integration QA	Use multi-hop information to answer questions	Full	Accuracy
T3.2 Single-Hop Fact QA	Answer questions based on local paragraphs	Partial	Accuracy
T4 Summarization & Synthesis	Generate abstract summary under given constraints		
T4.1 Global-Coverage Constrained Summary	Generate summary of full text	Full	SemSim, ROUGE-L
T4.2 Query-Focused Summary	Generate summary of specific subtopic	Partial	SemSim, ROUGE-L
T5 Attribution & Citation Alignment	Bind correct sources to generated text		
T5.1 Full-Sentence Citation Alignment	Citation alignment for all sentences	Full	F1
T5.2 Key-Statement Citation Alignment	Citation alignment for specified sentences	Partial	F1
T6 Aggregation & Clustering	Cluster and output statistics/examples/sort		
T6.1 Large-Scale Document Clustering	Return all category proportions	Full	SubEM
T6.2 Targeted Subset Cluster Identification	Return query category instances	Partial	F1
T6.3 Global Frequency Analysis	Count and sort global word frequency	Full	Pairwise Accuracy
T7 Consistency & Compliance Checking	Detect and locate contradictions/violations		
T7.1 Global Conflict & Inconsistency Localization	Locate contradictory segments in the full text	Full	F1
T7.2 Targeted Rule or Condition Violation Detection	Locate content that violates specific rules	Partial	F1
T7.3 Comprehensive Error & Anomaly Sweep	Locate spelling errors in the full text	Full	F1
T8 Structured & Numeric Reasoning	Numerical calculations in structured text		
T8.1 Structured Multi-Source Consistency Verification	Numerical computation in multi-source	Full	SubEM
T8.2 Single-Source Targeted Aggregation	Query computation in single-source	Partial	SubEM
T8.3 Long-Context Procedural State Tracking	Track entity state evolution	Full	SubEM
T9 Version & Code Diff Analysis	Compare changes in different text/code versions		
T9.1 Dependency-Aware Multi-Version Impact Analysis	Track dependency changes across versions	Full	F1
T9.2 Localized Interface Change Detection	Detect local version differences	Partial	F1
T10 Rule Induction & In-Context Learning	Summarize rules and make decisions on new samples		
T10.1 Large-Scale In-Context Rule Induction	Induce rules from the global context	Full	SubEM
T10.2 Targeted Example-Based Rule Induction	Induce rules from the targeted examples	Partial	SubEM
T11 Dialogue Memory & Long-Horizon Tracking	Track and respond to dialogue history		
T11.1 Long-Range Entity & Commitment Tracking	Track entity states across the global context	Full	Accuary
T11.2 Short-Range Reference Resolution & State Query	Resolve references and states in local context	Partial	Accuary

Table 2: Task definitions of LongBench-Pro.

aligned with a target task definition, including (i) questions, (ii) reference answers, and (iii) the corresponding design rationales and solution processes to support later verification.

Subsequently, human annotators critically evaluate the model-generated content, focusing primarily on the following aspects:

(1) Verify task alignment and context requirement based on the provided design rationale;

(2) Validate answer correctness using the accompanying solution processes;

(3) Estimate difficulty using the responses of the five drafting models (a sample is considered challenging if any model answers incorrectly);

(4) Select the best sample that meets the criteria or can be minimally edited to do so; if none, move to the next document.

This workflow leverages models for scalable drafting and humans for rigorous verification, mitigating both human cognitive limitations at extreme lengths and model hallucinations. Each accepted sample is reviewed by a long-context expert; failed cases must be revised until they satisfy the criteria. Section 5.6 empirically evaluates the effectiveness of this construction strategy.

3.3 Question Standardization

Chain-of-Thought prompting (Wei et al., 2022) demonstrates that the thinking process enhances model performance. To systematically evaluate the upper bound of model capabilities, we construct two standardized prompt templates for each question: a non-thinking prompt and a thinking prompt. Each prompt includes a task description, output requirements, and an output example, uniformly instructing the model to present answer elements

line by line, following the identifier "[Answer]" for automated extraction and evaluation. The only difference between the two types of prompts is that the non-thinking prompt requires the model to answer directly. In contrast, the thinking prompt requires it to perform explicit step-by-step thinking before producing the final answer.

3.4 Answer Review

To ensure sample quality, we systematically review all samples. Annotation experience shows that human annotators are more reliable in judging the correctness of answer components, while models are better at generating diverse candidate components. Based on this complementarity, we first collect predictions from five advanced models for each sample. Then, we instruct annotators to examine each component of the original answer to ensure precision, followed by reviewing model predictions to improve recall. Two annotators independently verify each sample. Samples without detected issues are directly included in the benchmark. If either annotator identifies a potential problem, an additional long-context expert evaluates the sample and decides whether it requires reconstruction.

3.5 Difficulty Classification

To improve the benchmark's utility in real-world applications, we assign each sample a difficulty label defined from a model-centric perspective rather than subjective human ratings (Bai et al., 2025), which aligns more closely with the practical needs of contemporary LLM evaluation and provides a more natural foundation for the co-evolution of benchmarks and model capabilities. Concretely, we rank models by overall performance and partition them into three tiers (high/mid/low). Within each tier, we select five representative models that perform the best while covering diverse architectures, which reduces sensitivity to outliers and avoids the bias introduced by relying on a single model. This tiered design provides multiple decision boundaries for fine-grained difficulty labeling. The selected models refer to Appendix D.

On this basis, we divide the samples into four difficulty levels according to the performance of three groups of models, defined as follows:

- **Extreme**: samples that at most one high-performing model can answer correctly (for which a score greater than 0.65 on the summarization task is considered correct);
- **Hard**: after excluding Extreme samples, samples that at most one mid-performing model can answer correctly;
- **Moderate**: after further excluding Hard samples, samples that at most one low-performing model can answer correctly;
- **Easy**: the remaining samples are automatically assigned to this level.

This multi-tier progressive approach yields fine-grained difficulty labels aligned with model capabilities and provides a unified, scalable framework for analyzing cross-difficulty performance.

4 Data Statistics and Validation of LongBench Pro

We construct LongBench Pro with a balanced design: 5 samples for each combination of 25 secondary tasks, 2 languages, and 6 length buckets, resulting in 1,500 samples in total. Appendix E reports the distributions of text type, language, context requirement, length, and difficulty, as well as the per-task composition. To validate sample quality, we uniformly select 300 samples across secondary tasks, languages, and lengths and audit (i) attribute correctness (whether language, length, secondary task, and context requirement are all correct) and (ii) answer correctness (whether the answer is fully correct). Among the 300 samples, the attribute correctness reaches 99.3%, and the answer correctness reaches 97.3%, with problematic samples exhibiting only minor deviations (impacting the overall score by only 0.96), demonstrating the high quality of the benchmark samples.

5 Evaluation

5.1 Evaluation Settings

Evaluation Metrics: Table 2 details task-specific metrics, all metrics have a value range of $[0, 1]$. We report the average score over all samples and multiply by 100. For T4 (summarization), we combine semantic similarity (SemSim) and ROUGE-L weighted equally to balance semantic faithfulness and coverage, computing each against three reference answers and taking the maximum to ensure best-match consistency. The design of the remaining metrics is detailed in Appendix F.1.

$$\begin{aligned}\text{Score}_{\text{summary}} &= 0.5 \cdot \max_i \text{SemSim}(S_{\text{gen}}, S_{\text{ref}_i}) \\ &\quad + 0.5 \cdot \max_i \text{ROUGE-L}(S_{\text{gen}}, S_{\text{ref}_i})\end{aligned} \tag{1}$$

Evaluated Models: We evaluate 46 widely discussed models that differ in transparency, thinking mode, size, architecture, and context length, with the goal of comprehensively assessing the long-context performance of current LLMs. Details of model selection can be found in Appendix F.2.

Inference Settings: We run each model three times with its default inference parameters and report the average score. Non-thinking and thinking scores are computed using different prompt templates based on the model's thinking mode. Samples exceeding the context length are truncated in the middle. Detailed inference settings are in Appendix F.3. Thinking scores are reported by default.

5.2 General Performance

Table 3 summarizes the general performance of 46 models on LongBench Pro (gray-shaded cells are thinking scores). We observe a clear stratification in overall long-context performance. The top three models are Gemini-2.5-Pro (73.42), GPT-5 (72.61), and Claude-4-Sonnet (69.87). Among open-source models, DeepSeek-V3.2 (67.82) and Qwen3-235B-A22B-Thinking-2507 (66.97) are the strongest, narrowing the gap to the best closed-source model to within 6 points.

Based on an in-depth analysis of the evaluation results in Table 3, we draw the following insights:

(1) Long-Context Optimization Outperforms Model Size Scaling. While scaling laws hold (Qwen3 supports 32k natively, 4B→32B: 40.82→51.12 with diminishing returns), optimizing context length proves more effective. Qwen3-4B-Instruct-2507 (256k) at 45.68 surpasses Qwen3-8B (44.34), and Qwen3-30B-A3B-Instruct-2507 (256k) at 54.52 outperforms the larger Qwen3-32B (51.12), demonstrating that extending effective context length surpasses parameter scaling for long-context performance.

(2) Discrepancy between Claimed and Effective Context Length. Claimed context length poorly correlates with performance. MiniMax-Text-01 (4M claimed) scores only 45.00, underperforming most 128k models, revealing that input capacity does not ensure effective information utilization. Claimed length reflects input capacity; effective length reflects actual comprehension and reasoning capability.

(3) Uneven Cross-Lingual Long-Context Capabilities. Models exhibit language-specific performance: GPT, Claude, Mistral, and Llama excel in English; GLM, Kimi, and MiniMax excel in Chinese, indicating insufficient language alignment. However, high-performing models (e.g., DeepSeek-V3.2, Qwen3-235B-A22B-Thinking-2507) narrow this gap through stronger cross-lingual representations, suggesting future convergence toward genuine cross-lingual consistency.

(4) Extreme Difficulty Reveals True Capability Gaps. Open-source and closed-source models perform comparably on Easy samples (GPT-5: 85.23 vs. DeepSeek-V3.2: 85.02), but gaps widen dramatically on Extreme samples (Gemini-2.5-Pro: 50.77, GPT-5: 48.37, DeepSeek-V3.2: 44.27, Qwen3-235B-A22B-Thinking-2507: 43.39). Thinking gains also diminish: Claude-4-Sonnet gains +15.36 on Easy but only +4.13 on Extreme; Gemini-2.5-Flash gains +13.27 on Easy but only +3.13 on Extreme. This reveals that Extreme samples test combined memory, integration, and reasoning capabilities, where current models have substantial room for improvement.

(5) "Thinking" Paradigm Becomes a Key Breakthrough. Nearly all models benefit from thinking: Gemini-2.5-Flash (55.92→67.41) and Qwen3-235B-A22B-Instruct-2507 (52.51→63.77). Notably, Qwen3-4B with thinking (40.82) surpasses Qwen3-32B without (40.28), bridging size gaps. This indicates thinking is crucial for capturing cross-paragraph connections in long-context information retrieval and aggregation.

(6) Native vs. Prompted Thinking Gap. Thinking or mixed-thinking models achieve substantial gains through enabling thinking: Claude-4-Sonnet (+13.80), DeepSeek-V3.2 (+16.15). Conversely, instruct models gain minimally or degrade: Llama-3.1-405B-Instruct (+0.59), Gemma-3-12B-It (-0.24), Llama-3.1-8B-Instruct (-1.03). This demonstrates thinking is a fundamental post-training paradigm shift, not prompt engineering—native thinking capability is key for long-context performance.

(7) Mixed-Thinking Models Exhibit Pareto-Optimal. Mixed-thinking models achieve Pareto optimality, balancing fast response and deep reasoning. They maintain robust baselines when thinking is disabled and match or exceed thinking models when thinking is enabled (e.g., Gemini-2.5-Flash approaches Gemini-2.5-Pro; DeepSeek-V3.2 outperforms DeepSeek-R1). This dynamic adaptability positions mixed-thinking as the most competitive future paradigm.

Model	Model Type	Context Length	Overall		Language				Difficulty							
					English		Chinese		Extreme		Hard		Moderate		Easy	
Gemini-2.5-Pro	Thinking	1M	-	73.42	-	72.35	-	74.49	-	50.77	-	81.03	-	81.98	-	84.40
Gemini-2.5-Flash	Mixed	1M	55.92	67.41	55.29	67.22	56.54	67.59	44.26	47.39	57.87	72.19	53.99	72.39	66.55	79.82
Gemma-3-27B-It	Instruct	128k	36.14	37.34	37.46	40.89	34.81	33.78	30.22	27.78	33.20	30.56	25.04	24.53	49.96	57.81
Gemma-3-12B-It	Instruct	128k	32.16	31.92	33.03	34.43	31.28	29.41	26.44	25.74	30.43	28.02	23.39	22.61	43.66	45.48
Gemma-3-4B-It	Instruct	128k	21.76	21.20	22.63	23.28	20.89	19.12	19.31	18.72	20.70	19.87	15.82	13.85	28.18	28.66
GPT-5	Thinking	272k	-	72.61	-	73.24	-	71.97	-	48.37	-	78.74	-	82.31	-	85.23
GPT-4o	Instruct	128k	46.67	49.44	47.67	52.61	45.66	46.26	36.30	34.39	44.88	41.35	43.03	43.07	59.38	71.84
GPT-OSS-120B	Thinking	128k	-	52.61	-	54.67	-	50.54	-	35.4	-	44.97	-	50.66	-	74.06
GPT-OSS-20B	Thinking	128k	-	44.66	-	47.83	-	41.49	-	31.59	-	35.89	-	39.33	-	65.05
Claude-4-Sonnet	Mixed	1M	56.07	69.87	57.14	71.09	54.99	68.65	42.92	47.05	57.57	74.72	53.96	76.58	68.42	83.78
Claude-3.7-Sonnet	Mixed	200k	51.45	59.66	51.89	60.49	51.00	58.84	37.31	40.07	47.29	56.58	48.38	61.56	68.69	78.26
DeepSeek-V3.2	Mixed	160k	51.67	67.82	50.61	67.89	52.73	67.75	40.45	44.27	51.63	67.73	51.36	75.08	62.12	85.02
DeepSeek-V3.1	Mixed	128k	51.39	66.22	50.35	66.17	52.42	66.26	41.07	42.68	49.29	62.22	48.80	73.53	63.61	85.72
DeepSeek-R1-0528	Thinking	128k	-	61.89	-	59.90	-	63.89	-	41.49	-	53.68	-	66.53	-	82.67
DeepSeek-R1	Thinking	128k	-	60.07	-	62.00	-	58.13	-	40.76	-	53.39	-	58.83	-	82.44
DeepSeek-V3-0324	Instruct	128k	51.70	56.71	51.62	58.14	51.78	55.27	40.40	38.69	48.30	46.20	49.68	57.14	65.26	79.20
Qwen3-235B-A22B-Thinking-2507	Thinking	256k	-	66.97	-	66.83	-	67.12	-	43.39	-	67.10	-	75.12	-	83.55
Qwen3-235B-A22B-Instruct-2507	Instruct	256k	52.51	63.77	52.22	63.88	52.80	63.65	42.07	43.24	52.34	58.60	53.14	68.15	61.76	82.98
Qwen3-Next-80B-A3B-Thinking	Thinking	256k	-	63.95	-	62.91	-	64.99	-	42.47	-	61.46	-	69.23	-	81.90
Qwen3-Next-80B-A3B-Instruct	Instruct	256k	51.54	60.76	50.39	60.30	52.69	61.22	39.39	40.47	49.93	54.74	48.77	64.16	65.25	80.84
Qwen3-30B-A3B-Thinking-2507	Thinking	256k	-	59.68	-	60.14	-	59.22	-	40.47	-	52.76	-	62.55	-	79.64
Qwen3-30B-A3B-Instruct-2507	Instruct	256k	43.84	54.52	43.17	55.55	44.5	53.49	35.44	37.05	41.32	44.04	39.04	56.47	55.89	75.59
Qwen3-4B-Thinking-2507	Thinking	256k	-	50.10	-	49.81	-	50.39	-	35.31	-	40.99	-	47.66	-	70.53
Qwen3-4B-Instruct-2507	Instruct	256k	36.78	45.68	35.70	46.27	37.85	45.10	30.29	31.09	33.94	36.96	30.21	39.69	48.33	67.82
Qwen3-32B	Mixed	128k	40.28	51.12	39.78	52.24	40.77	50.01	32.56	36.45	38.61	42.24	34.30	46.18	51.90	72.80
Qwen3-14B	Mixed	128k	37.11	47.14	36.61	50.53	37.61	43.75	31.13	33.66	35.34	38.41	29.07	39.03	48.44	69.55
Qwen3-8B	Mixed	128k	33.41	44.34	33.04	44.60	33.79	44.08	29.99	33.50	31.09	37.10	25.20	30.16	42.86	67.08
Qwen3-4B	Mixed	128k	31.26	40.82	31.60	41.94	30.92	39.70	27.20	30.69	30.10	34.07	23.33	31.27	40.43	59.85
Qwen2.5-72B-Instruct	Instruct	128k	39.64	44.09	39.48	44.99	39.79	43.18	32.36	31.71	35.90	36.45	31.36	31.03	53.48	67.80
GLM-4.6	Mixed	198k	45.85	58.21	45.64	56.50	46.07	59.92	37.30	38.88	43.07	48.92	40.05	60.95	58.82	79.78
GLM-4.5	Mixed	128k	43.04	55.48	43.05	53.57	43.02	57.39	35.06	37.94	40.21	47.38	36.92	55.13	55.68	76.55
Kimi-K2-Instruct-0905	Instruct	256k	50.09	55.53	49.90	56.96	50.29	54.10	39.61	38.25	43.43	43.75	49.05	57.33	64.92	77.29
MiniMax-M2	Thinking	192k	-	53.21	-	52.87	-	53.55	-	34.98	-	42.58	-	59.92	-	72.20
MiniMax-Text-01	Instruct	4M	41.14	45.00	40.21	44.17	42.06	45.82	33.57	33.78	38.67	38.02	38.23	40.82	51.26	61.92
Ministral-3-14B-Instruct-2512	Instruct	256k	40.14	45.80	39.71	47.75	40.56	43.85	33.85	31.66	35.04	37.48	34.02	39.35	52.60	67.56
Ministral-3-8B-Instruct-2512	Instruct	256k	37.80	44.46	36.61	46.17	39.00	42.75	31.88	31.86	32.11	34.99	31.73	35.26	50.27	67.14
Ministral-3-3B-Instruct-2512	Instruct	256k	30.18	34.54	27.75	36.42	32.61	32.66	25.97	26.70	28.57	30.23	23.80	25.60	38.81	49.65
Magistral-Small-2509	Thinking	128k	-	38.40	-	40.40	-	36.40	-	30.52	-	32.92	-	29.44	-	54.25
Mistral-Small-3.2-24B-Instruct-2506	Instruct	128k	37.32	39.87	39.23	42.56	35.41	37.18	31.79	29.77	33.73	31.76	27.42	27.74	50.45	61.22
Mistral-Large-Instruct-2411	Instruct	128k	31.69	36.25	33.10	39.14	30.28	33.36	27.39	28.65	29.88	29.42	23.42	25.62	41.66	53.65
Ministral-8B-Instruct-2410	Instruct	128k	17.56	14.43	18.65	16.53	16.47	12.33	17.83	15.06	18.61	13.98	12.26	9.89	19.86	16.84
Llama-3.1-405B-Instruct	Instruct	128k	40.07	40.66	42.03	44.46	38.11	36.86	33.44	29.81	35.51	34.09	29.07	29.22	55.45	61.36
Llama-3.3-70B-Instruct	Instruct	128k	31.89	33.69	35.12	39.15	28.66	28.23	26.53	24.32	29.07	28.59	22.61	22.61	44.04	51.94
Llama-3.1-70B-Instruct	Instruct	128k	31.53	32.12	35.10	36.85	27.96	27.40	26.22	23.93	28.86	28.04	21.46	21.44	44.02	48.46
Llama-3.1-8B-Instruct	Instruct	128k	21.09	20.06	24.28	25.40	17.91	14.71	21.00	19.68	21.22	17.99	13.82	12.32	25.47	26.28
Llama-3.2-3B-Instruct	Instruct	128k	15.71	12.58	20.90	16.45	10.51	8.71	16.63	15.57	15.01	10.48	10.37	7.17	18.49	14.35

Table 3: General performance on LongBench Pro. Gray-shaded cells represent thinking scores. The best three performance results are highlighted using **red** (1st), **green** (2nd), and **blue** (3rd) font colors, respectively.

5.3 Comparison Across Task Dimension

Figure 8 shows the performance of mainstream LLMs across tasks. We make the following observations:

(1) Significant gap between retrieval and aggregation. While models excel at basic information retrieval (T1) and sequence reconstruction (T2) (with average scores above 80), performance drops sharply on semantic aggregation (T6, average 57.72), requiring complex information integration. This indicates that despite precise "needle-in-a-haystack" localization, models still face challenges in semantically aggregating dispersed information across long contexts.

(2) Imbalance in forward and backward inference. Models perform well on evidence retrieval (T5, average 63.47) but comparatively lower on question answering (T3) and summarization (T4), tasks requiring forward derivation from detailed documents (average below 55). This suggests stronger backward alignment from outcomes to evidence, while forward generation remains more susceptible to document complexity.

(3) Logical reasoning and consistency as high-level bottlenecks. Most models perform moderately on logical reasoning (T8–T10, average around 60), with reasoning-enhanced models showing significant gains. However, consistency maintenance (T7, T11, average below 49) remains challenging, exposing limitations in sustaining global states over very long sequences.

5.4 Comparison Across Length Dimension

Figure 3 shows that most models exhibit declining performance as sample length increases. However, Gemini-2.5-Pro demonstrates remarkable length in-

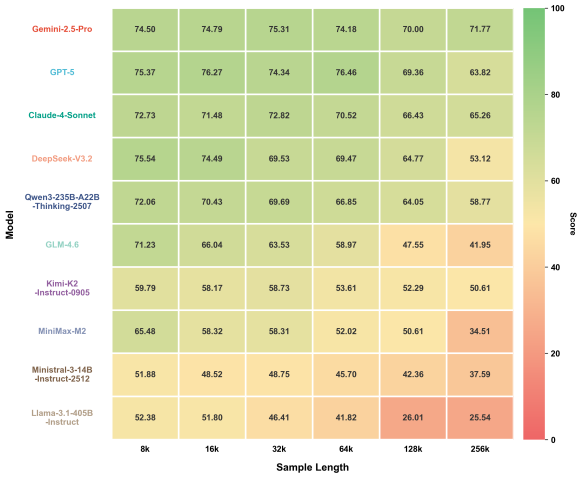


Figure 3: Performance across different sample lengths.

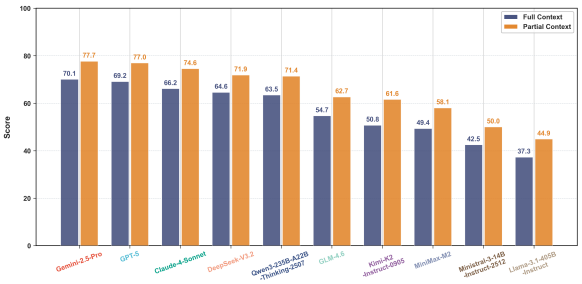


Figure 4: Performance across different context requirements.

sensitivity, maintaining similar scores at 8k (74.50) and 256k (71.77). This phenomenon indicates that, within the range of 256k in length, for current state-of-the-art long-context models, merely increasing sample length to stress-test model performance has reached a point of saturation. The current bottleneck lies not in reading capacity but in handling long-range dependencies and complex logical relationships, shifting evaluation focus from "how much can it read" to "how deeply can it comprehend."

5.5 Comparison Across Context Requirement Dimension

Figure 4 shows clear performance stratification under varying context requirements: all models perform substantially better on Partial tasks (localization) than Full tasks (integration), with performance drops of 7.32 to 10.84 points. This indicates that while current models excel at local information retrieval, they face notable limitations in cross-segment information association and holistic reasoning.

5.6 Comparison Across Construction Strategies

We construct samples from 50 uniformly sampled documents using three strategies: human-only, model-only, and human-model collaboration (ours). As shown in Figure 5 (a), time cost increases exponentially for human-only, remains consistently low for model-only, and grows slowly for human-model collaboration, demonstrating the significant efficiency improvements introduced by model participation.

For quality assessment, we employ a five-dimensional framework (scored 0-1 by three experts); results show human-model collaboration yields the highest quality (0.9609±0.0415), surpassing both human-only (0.9484±0.0450) and model-only (0.8964±0.0536), with strong inter-rater reliability (Fleiss' Kappa=0.76). In Figure 5 (b), all strategies achieve high marks in task alignment, context alignment, and authenticity—attributable to our clear definitions and real-world texts. However, model-only is disadvantaged in difficulty (due to lack of human filtering) and correctness (because of hallucinations), while human-only falls short in correctness by occasionally omitting answer components. Human-model collaboration remedies both types of deficiencies, attaining optimal difficulty and correctness. For further details, see Appendix G.3.

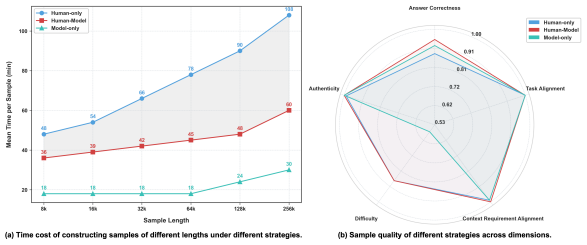


Figure 5: Comparison of sample construction strategies.

6 Conclusion

In this work, we introduce LongBench Pro, a realistic and comprehensive bilingual benchmark for long-context evaluation. We adopt a human–model collaborative sample construction strategy that effectively mitigates the trade-off between quality and cost. We further evaluate 46 representative long-context large language models on LongBench Pro and provide analyses across task, length, context requirement, difficulty, and language settings.

Limitations

As task length and complexity continue to grow, even human-model collaborative construction can face a tension between verification accuracy and production efficiency. We are exploring a recursive critique scheme ("Critique-of-Critique") to recursively and progressively decompose verification into easier subproblems that are tractable for human annotators.

Ethical Considerations

All data used in this study are obtained from publicly available sources and undergo manual screening to mitigate potential risks, including privacy breaches, copyright issues, and inappropriate content. All annotators participate voluntarily in the sample construction and annotation process and receive fair market compensation. Clear instructions are provided for annotation tasks, and working hours are reasonably managed to minimize labor-related risks. To prevent potential misuse, the benchmark is released under a research-only license and requires users to adhere to responsible use guidelines. We continuously monitor community feedback and promptly review and address any ethical or societal concerns that arise.

References

Meta AI. 2024. Introducing meta llama 3.1: Our most capable models to date. https://ai.meta.com/blog/meta-llama-3-1/.

Mistral AI. 2025. Introducing mistral 3. https://mistral.ai/news/mistral-3.

Chenxin An, Shansan Gong, Ming Zhong, Xingjian Zhao, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. 2023. L-Eval: Instituting standardized evaluation for long context language models. *arXiv preprint arXiv:2307.11088*.

Anthropic. 2025. Introducing claude 4. https://www.anthropic.com/news/claude-4.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and 1 others. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, and 1 others. 2025. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3639–3664.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, and 108 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

OpenAI. 2025. Introducing gpt-5.

OpenAI. 2025. Openai mrcr: Long context multiple needle in a haystack benchmark. https://huggingface.co/datasets/openai/mrcr. Dataset, MIT License, accessed 2025-12-19.

Zexuan Qiu, Jingjing Li, Shijue Huang, Xiaoqi Jiao, Wanjun Zhong, and Irwin King. 2024. Clongeval: A chinese benchmark for evaluating long-context large language models. *arXiv preprint arXiv:2403.03514*.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. 2023. Zeroscrolls: A zero-shot benchmark for long text understanding. *arXiv preprint arXiv:2305.14196*.

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. 2022. Scrolls: Standardized comparison over long language sequences. *arXiv preprint arXiv:2201.03533*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025a. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, and 1 others. 2025b. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*.

MiniMax Team. 2025. Minimax m2 & agent: Ingenious in simplicity. https://minimaxi.com/news/minimax-m2.

Chonghua Wang, Haodong Duan, Songyang Zhang, Dahua Lin, and Kai Chen. 2024. Ada-L-Eval: Evaluating long-context LLMs with length-adaptable benchmarks. *arXiv preprint arXiv:2404.06480*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. 2024. Helmet: How to evaluate long-context language models effectively and thoroughly. *arXiv preprint arXiv:2410.02694*.

Zhipu AI (Z.ai). 2025. Glm-4.6: Advanced agentic, reasoning and coding capabilities. https://z.ai/blog/glm-4.6.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and 1 others. 2024. ∞bench: Extending long context evaluation beyond 100k tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15262–15277.

Yang Zhou, Hongyi Liu, Zhuoming Chen, Yuandong Tian, and Beidi Chen. 2025. Gsm-infinite: How do your llms behave over infinitely increasing context length and reasoning complexity? *arXiv preprint arXiv:2502.05252*.

A Task Mapping

Figure 6 shows the task mapping between LongBench Pro and existing benchmarks, with LongBench Pro covering the full spectrum of long-context capabilities assessed by existing benchmarks.

B Task Definitions

T1 Retrieval & Ranking

Retrieve content and rank most relevant first.

T1.1 Global Cohesive Retrieval

Retrieve full text and reorganize.

Context Requirement: Full

Metric: NDCG@k

Example:

> Retrieve all reviews that received 50 or more votes, and output the reviewerIDs in ascending order of vote value. Output the "`[Answer]`" identifier first, and then output reviewerIDs line by line, without any additional content.
>
> Output example:
> `[Answer]`
> A2AV7Q95QGPTO0
> A3NM0RAYSL6PA8
> A1C9C1QOQB94RT
> A1B80MVU7ZODF9

T1.2 Key-Snippet Retrieval

Locate target fragment in specified paragraph.

Context Requirement: Partial

Metric: NDCG@k

Example:

> From the subset of articles written by "Dr. Seuss" (for which an artificial ID field has been added): Retrieve articles in the category "Life" and sort them by popularity from lowest to highest. Output the "`[Answer]`" identifier first, and then output the sorted reviewerID line by line, without any additional content.
>
> Output example:
> `[Answer]`
> ID1
> ID2
> ID3
> ID4

T2 Sequencing & Structure Reconstruction

Restore timeline or logical order.

T2.1 Global Timeline Reconstruction

Sort unordered events in the whole text.

Context Requirement: Full

Metric: Pairwise Accuracy

Example:

> This article is divided into nine parts, each part is preceded by "Part x", and the order is shuffled. Please sort these nine parts in the correct order of events. Output the "`[Answer]`" identifier first, and then output correct sequence of part numbers line by line, without any additional content.
>
> Output example:
> `[Answer]`
> Part 5
> Part 1
> Part 9
> Part 3
> Part 7
> Part 2
> Part 4
> Part 8
> Part 6

T2.2 Local Causal Chain Sorting

Sort content in a specific paragraph.

Context Requirement: Partial

Metric: Pairwise Accuracy

Example:

> The "SELECTED PARAGRAPH" of the article is out of order. Please reorder this paragraph according to the original text and options. Output the "`[Answer]`" identifier first, and then output the sorted option letters line by line, without any additional content.
>
> Output example:
> `[Answer]`
> A
> B
> C
> D
> E

T3 Evidence-Grounded QA

Answer fact/reasoning questions based on evidence.

T3.1 Multi-Doc Integration QA

Use multi-hop information to answer questions.

Context Requirement: Full

Metric: Accuracy

Example:

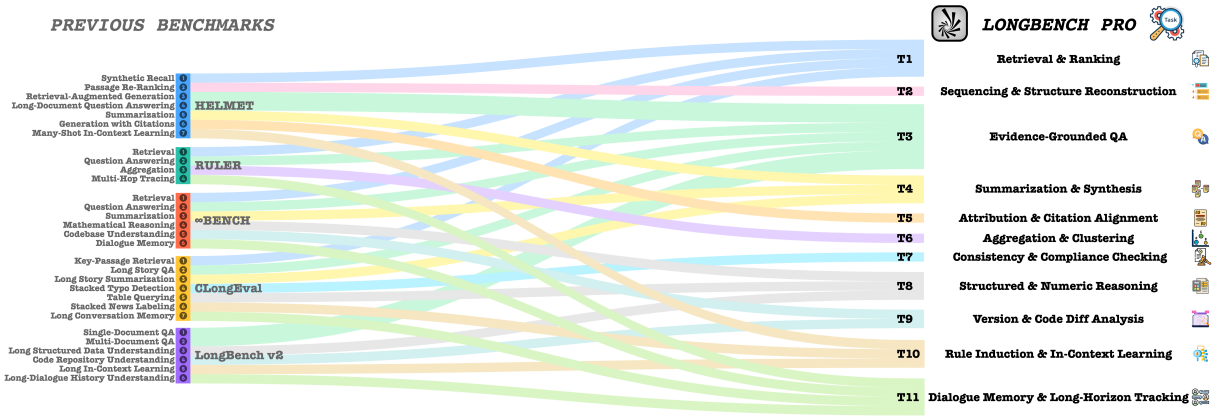


Figure 6: Task mapping between LongBench Pro and existing benchmarks.

On their way to Egloshayle, from the conversation of the passengers on the coach, it can be inferred that the characteristics of Mr. Alan Torrington do not include the following? Output the "[Answer]" identifier first, and then output the answer option letter (A/B/C/D), without any additional content.

A. He was rumored by some to be a reclusive person, with a habit of wandering near the rocks at night, and was even falsely labeled as a "mysterious wizard."
B. His personality was in stark contrast to that of his brother Oscar Torrington, and he was considered to be a miser, with rumors that he hid his wealth in secret places.
C. He was in poor health, and his trip to Switzerland with his brother was aimed at improving his health, as he usually required someone to take care of his daily life.
D. He had once placed a light in the tower window to mislead ships into running aground in order to acquire their cargo, becoming a figure associated with the local history of "wrecking."

Output example:
[Answer]
C

T3.2 Single-Hop Fact QA

Answer questions based on local paragraphs.
Context Requirement: Partial
Metric: Accuracy
Example:

What are all the topic categories listed in the paper "Delta Activities: A Representation for Finetuned Large Language Models" (arXiv: 2509.04442)? Output the "[Answer]" identifier first, and then output the answer option letter (A/B/C/D), without any additional content.

A. Machine Learning (cs.LG); Computation and Language (cs.CL); Artificial Intelligence (cs.AI)
B. Artificial Intelligence (cs.AI); Computation and Language (cs.CL); Information Retrieval (cs.IR)
C. Machine Learning (cs.LG); Artificial Intelligence (cs.AI); Computation and Language (cs.CL); Information Retrieval (cs.IR)
D. Machine Learning (cs.LG); Artificial Intelligence (cs.AI); Information Retrieval (cs.IR); Sound (cs.SD)

Output example:
[Answer]
A

T4 Summarization & Synthesis

Generate summary under given constraints.

T4.1 Global-Coverage Constrained Summary

Generate summary of full text.
Context Requirement: Full
Metric: 0.5*max(SemSim)+0.5*max(ROUGE-L)
Example:

The above are multiple chapters of a novel, and based on the given text, summarize and generalize the entire content, with a requirement of no more than 100 words. Output the "[Answer]" identifier first, and then output the summary, without any additional content.

Output example:
[Answer]
your summary

T4.2 Query-Focused Summary

Generate summary of specific subtopic.

Context Requirement: Partial

Metric: 0.5*max(SemSim)+0.5*max(ROUGE-L)

Example:

> Summarize the plot of the absurd victory celebration only in the first section ($\leq$ 200 words). Output the "`[Answer]`" identifier first, and then output the summary, without any additional content.
>
> Output example:
> `[Answer]`
> your summary

T5 Attribution & Citation Alignment

Bind correct sources to generated text.

T5.1 Full-Sentence Citation Alignment

Citation alignment for all sentences.

Context Requirement: Full

Metric: F1

Example:

> You will be provided with a summary sentence. Your task is to identify the original Part number(s) from the provided text that fully support this sentence. Output the "`[Answer]`" identifier first, and then output the cited content (format as "Part xx") line by line, without any additional content.
>
> Generated Summary:
> After taking a leave of absence post the 1984 Macintosh launch (to recharge and avoid supervisor Bob Belleville), Andy Hertzfeld clashed with Jobs first. Over a denied Mac team bonus (which Jobs initially blamed on Belleville. Before relenting, leaving Hertzfeld upset) and later when Jobs dismissed his. Concerns about the Mac software team's low morale and Burrell's frustration. As out of touch, insisting the team was doing well.
>
> Output example:
> `[Answer]`
> Part 1
> Part 12
> Part 15

T5.2 Key-Statement Citation Alignment

Citation alignment for specified sentences.

Context Requirement: Partial

Metric: F1

Example:

> You will see a generated 3-sentence summary. Identify and cite the original source location for Sentence 2 only, using the paragraph identifiers S1–S120 from the source text. Output the "`[Answer]`" identifier first, and then output the paragraph identifier(s) line by line, without any additional content.
>
> Generated summary:
> Sentence 1: During Danielle Mitterrand's factory tour, she pressed on labor conditions—overtime and vacation—while Jobs, increasingly annoyed, emphasizedautomation and just-in-time processes.
> Sentence 2: Jobs believed that Sculley did not understand business and was not suitable to manage the company, and designed a plan to remove him.
> Sentence 3: Driving back toward Cupertino, Jobs was ticketed for speeding near 100 mph and, unfazed, accelerated again—evidence, Rossmann said, that Jobs believed normal rules didn't apply to him.
>
> Output example:
> `[Answer]`
> S1
> S2

T6 Aggregation & Clustering

Cluster and output statistics/examples/sort.

T6.1 Large-Scale Document Clustering

Return all category proportions.

Context Requirement: Full

Metric: SubEM

Example:

> Cluster the 8 documents (IDs: A–H) by research methodology into exactly three clusters: Randomized trial/protocol; Quantitative observational; Qualitative. Output the "`[Answer]`" identifier first, and then output the three clusters in the format "ClusterName Proportion% (rounded to two decimal places)" line by line, without any additional content.
>
> Output example:
> `[Answer]`
> Randomized trial/protocol 25.00%
> Quantitative observational 25.00%
> Qualitative 50.00%

T6.2 Targeted Subset Cluster Identification

Return query category instances.

Context Requirement: Partial

Metric: F1

Example:

> Find the first three software reviews in the Software Reviews category. Output the "[Answer]" identifier first, and then output the answer reviewer IDs that meet the conditions line by line, without any additional content.
>
> Output example:
> [Answer]
> A22V1MD93T2FW9
> ACJT8MUCOLRFO
> A38NELQT98S4H8

T6.3 Global Frequency Analysis

Count and sort global word frequency.

Context Requirement: Full

Metric: Pairwise Accuracy

Example:

> Sort the given terms in descending order by the number of times they appear in the above text. Output the "[Answer]" identifier first, and then output the sorted terms line by line, without any additional content.
>
> Given terms: would, this, that, went, have
>
> Output example:
> [Answer]
> would
> this
> that
> went
> have

T7 Consistency & Compliance Checking

Detect and locate contradictions/violations.

T7.1 Global Conflict & Inconsistency Localization

Locate contradictory segments in the full text.

Context Requirement: Full

Metric: F1

Example:

> Compare the inconsistent articles between clauses A and B in two documents. Output the "[Answer]" identifier first, and then output all inconsistent articles line by line in the format "A[articleID] B[articleID]", without any additional content.
>
> Output example:
> [Answer]
> A1 B2
> A2 B3

T7.2 Targeted Rule or Condition Violation Detection

Locate content that violates specific rules.

Context Requirement: Partial

Metric: F1

Example:

> Read and list which terms of Enzuzo, Inc. Privacy Policy were violated in the first article. Output the "[Answer]" identifier first, and then output all the violated term IDs line by line, without any additional content.
>
> Output example:
> [Answer]
> 5
> 6

T7.3 Comprehensive Error & Anomaly Sweep

Locate spelling errors in the full text.

Context Requirement: Full

Metric: F1

Example:

> Find and list all misspelled English words in a given complete text file. Output the "[Answer]" identifier first, and then output all misspelled English words line by line, without any additional content.
>
> Output example:
> [Answer]
> beleive
> freind
> tommorow

T8 Structured & Numeric Reasoning

Numerical calculations in structured text.

T8.1 Structured Multi-Source Consistency Verification

Numerical computation in multi-source.

Context Requirement: Full

Metric: SubEM

Example:

> Above are two tables. Please use the following formula to determine if there are any issues after merging the two tables (i.e., whether the sales amount calculated using the formula for Table 1 is consistent with the sales amount in Table 2).

If there are no issues, return "No Error". If there are any issues, return the corresponding Invoice IDs. Output the "`[Answer]`" identifier first, and then output "No Error" or output corresponding Invoice IDs line by line, without any additional content.

Formula: Sales Amount = Unit Price × Quantity + 5% Tax

Output example:
`[Answer]`
No Error/226 31 3081

T8.2 Single-Source Targeted Aggregation

Query computation in single-source.
Context Requirement: Partial
Metric: SubEM
Example:

According to the table "Share repurchases for the fourth quarter of fiscal year 2023", calculate the percentage increase in the "Total Number of Shares Purchased as Part of Publicly Announced Plans or Programs" for June compared to April and May. Output the "`[Answer]`" identifier first, and then output percentages (with one decimal place retained) line by line, without any additional content.

Output example:
`[Answer]`
5.5%
-2.2%

T8.3 Long-Context Procedural State Tracking

Track entity state evolution.
Context Requirement: Full
Metric: F1
Example:

Find and list all misspelled English words in a given complete text file. Output the "`[Answer]`" identifier first, and then output all misspelled English words line by line, without any additional content.

Output example:
`[Answer]`
beleive
freind
tommorow

T9 Version & Code Diff Analysis

Compare changes in different text/code versions.

T9.1 Dependency-Aware Multi-Version Impact Analysis

Track dependency changes across versions.
Context Requirement: Full
Metric: F1
Example:

Based on the two provided API documents for hadoop-core (version 0.20.0 and 0.21.0), identify all methods within the "org.apache.hadoop.fs.FileSystem" class that were not deprecated in version 0.20.0 but became deprecated in version 0.21.0. Use the method signature as the identifier in the format "MethodName(ParameterTypes)". If a method has no parameters, use "MethodName()". For multiple parameters, use a comma-separated list of fully qualified types. Output the "`[Answer]`" identifier first, and then output the identifiers line by line, without any additional content.

Output example:
`[Answer]`
MethodA(java.lang.String)
MethodB()
MethodC(int, java.lang.String)

T9.2 Localized Interface Change Detection

Detect local version differences.
Context Requirement: Partial
Metric: F1
Example:

Identify the fields within the "Lesson" resource model that were either renamed, removed due to architectural changes, or refactored into a more complex structure when evolving from V1.0 to V2.0. List these fields using the format "Lesson Model: [V2.0 Field Name]" for renamed/refactored fields and "Lesson Model: [V1.0 Field Name]" for fields that were entirely removed. Output the "`[Answer]`" identifier first, and then output fields line by line, without any additional content.

Output example:
`[Answer]`
Lesson Model: key
Lesson Model: seq_length

T10 Rule Induction & In-Context Learning

Summarize rules and decide on new samples.

T10.1 Large-Scale In-Context Rule Induction

Induce rules from the global context.

Context Requirement: Full
Metric: SubEM
Example:

> You are a script formatter. Based on the full range of script formatting conventions demonstrated in the provided text, reformat the following raw script snippet. Pay close attention to scene headings, character names, dialogue parenthetical actions, and transitions. Output the "`[Answer]`" identifier first, and then output the fully formatted script snippet, without any additional content.
>
> Raw Snippet:
> interior, lab - day
> Dr. Banner (looking at monitor): the readings are stable now. Tony Stark: That's what I like to hear.
> (Tony smiles)
> cut to:
> exterior, city rooftop - night
> A figure (hooded) stands looking at the skyline.
>
> Output example:
> `[Answer]`
> your answer

T10.2 Targeted Example-Based Rule Induction

Induce rules from the targeted examples.
Context Requirement: Partial
Metric: SubEM
Example:

> Answer the following questions based on the content of the case in Part 1 of the provided document: UKCo is a private limited company specializing in cross-border private equity investments. In 2022, UKCo signed a Joint Investment Agreement with three EU investment institutions (A, B, C), agreeing to jointly establish a European infrastructure cooperation plan to invest in EU new energy infrastructure projects. In 2024, the European Infrastructure Cooperation Program fell into a debt crisis due to project defaults, owing 8 million euros to supplier Company D (a German company). At the same time, A, B, and C found that UKCo had misappropriated funds from the European Infrastructure Cooperation Scheme in its management, so they applied to the English Commercial Court: Should the English court support A, B, and C's application for compulsory liquidation of the European Infrastructure Cooperation Scheme as a compulsory liquidation? Output the "`[Answer]`" identifier first, and then output answer. The answer must be a fixed value ("support/don't support"), and without any additional content.
>
> Output example:
> `[Answer]`
> support

T11 Dialogue Memory & Long-Horizon Tracking

Track and respond to dialogue history.

T11.1 Long-Range Entity & Commitment Tracking

Track entity states across the global context.
Context Requirement: Full
Metric: Accuary
Example:

> Based on the conversation between the user and the model above, which of the following options best fits the content in the text? Output the "`[Answer]`" identifier first, and then output A/B/C/D/E, without any additional content.
>
> A. About the book "Ship of Theseus", Based on the conversation, here is a simple and insightful summary of the book "S." / "Ship of Theseus": "S." is a unique, single level reading experience that casts the reader as a detective.It is a book within a book, containing.
> B. About the book "Piranesi", Based on the conversation, here is a summary of the discussion about Susanna Clarke's novel "Piranesi": A signature style of the author, Susanna Clarke, She uses authentic academic citations, which lends the story a remarkable sense of authenticity and depth while also concealing clues.
> C. About the movie "Lobster", Based on the conversation, here is a summary of the analysis of the movie "The Lobster": The discussion analyzes Yorgos Lanthimos's 2015 film "The Lobster," identifying its core premise-a society where single people must find a partner within a month or become an animal-as a satirical critique of modern society's pressure to be in a relationship.
> D. About "Dark City", The user and model discuss the 1999 Alex Proyas film "Dark City", agreeing it is an underrated sci-fi classic often overshadowed by "The Matrix". They identify

its core premise: a city perpetually trapped in night where alien "Strangers" experiment on humans by altering their memories and physical reality to understand the human soul, which they themselves lack.
E. "The Ship of Theseus", The Joy of Collaboration: A central theme is that the best way to enjoy these complex works is through collaboration. The model encourages the user to solve puzzles oneself, suggesting they can divide tasks and that debating theories will lead to deeper insights. This is supported by real-world examples of fan communities and is also applied to finding hidden details (like changing tie patterns) in the film "Predestination" ("Former Destination").

Output example:
`[Answer]`
B

T11.2 Short-Range Reference Resolution & State Query

Resolve references and states in local context.
Context Requirement: Partial
Metric: Accuary
Example:

In the scene in the elegant dining room where Cobb and Arthur are pitching their services to Saito (pages 2-4), what specific reason does Arthur give for why a person's thoughts become vulnerable to theft in the dream state? Output the "`[Answer]`" identifier first, and then output the answer option letter (A/B/C/D), without any additional content.

A. Because a person can be trained to reveal their own secrets.
B. Because killing someone in a dream will not wake them up.
C. Because a person's conscious defenses are lowered.
D. Because an idea, once it takes hold, is impossible to eradicate.

Output example:
`[Answer]`
B

C Annotation Guidelines

C.1 Sample-Generation Prompt

Role Definition
You are a professional question-design specialist who:
• possesses solid capabilities in textual deconstruction and deep reading;
• has extensive cross-domain experience in exam item construction;
• demonstrates accurate command of the following domains: legal and regulatory texts, literature and creative writing, academic and professional materials, consulting and media content, history and culture, as well as general-interest texts;
• is familiar with each domain's technical terminology, logical systems, textual features, and core assessment points.

Task Objective
Given an input **long context** and its **task category (primary task + secondary task)**, you create **three questions** that strictly fulfill the task requirements (including **answers**, **design rationale**, and detailed **solution process**).

Task Procedure
Step 1: Language Identification
Determine the language of the input long text (e.g., Chinese/English). All subsequent content (questions, answers, explanations) must **fully use the same language** as the long context.
Step 2: Question Construction
2.1 Number of Questions
Based on the long context and the secondary task type, extract deep assessment points and generate **three questions** that satisfy the secondary task specifications.
2.2 Construction Requirements
• All questions must strictly satisfy the evaluation goals of the primary task, the required I/O format, and the definition of the secondary task.
• Question phrasing must follow the secondary task's example patterns.
• Answer formats must match the task examples.
• The three questions must exhibit clear differentiation, avoiding repeated formats or duplicate assessment points.
2.3 Sample Requirements

(1) Identifier Rules (ensuring answer uniqueness)
To avoid ambiguity caused by overlapping textual expressions, each question must apply identifiers:
① Natural identifiers (preferred). Use naturally occurring, unique elements in the document as identifiers, such as titles, section names, or explicit entities.
② Constructed identifiers. If natural identifiers are insufficient, create identifiers for document segments, e.g., DocumentX ParagraphX IDX D11. Requirements:
• Use only letters and numbers; do not include symbols (such as "- , _ .").
• Explicitly explain the numbering rule in the question.
• Require respondents to answer using the identifiers.
(2) Requirements for Multiple-Choice Questions (if applicable)
• Must include distractors based on the document.
• At least four options.
• The correct option must be verifiable in the text.
• Distractors must be reasonable and sufficiently plausible.
(3) Reference Answer (for summarization tasks only)
• Three results are required for semantic similarity evaluation; must be accurate and cover key points.
(4) Question-Design Rationale (required for every question)
You must explain:
• how the question satisfies the task type;
• which document information the question draws upon;
• why the answer is unique and verifiable.
(5) Solution and Evidence (required for every question)
You must provide:
• the complete, correct answer;
• detailed solution steps;
• clear citation of all textual evidence, with one-to-one correspondence.
(6) Unified Output Format
• All questions must require line-by-line answers to support automated evaluation.

2.4 Task Types

(1) Context Requirement Dimension
• Full: The context appears in fully scrambled order. A complete reading of the entire text is necessary to retrieve all content required for the question; missing any segment leads to an inaccurate answer.
• Partial: Since the relevant segments are contiguous and ordered, one only needs to locate the target segment; the rest is not essential.
(2) Specific Task Type
{Primary_Task_Definition}
{Secondary_Task_Definition, Context_Requ -irement, I/O_Specification, and Examples}

Step 3: Self-Check
After generating the three questions, you must conduct a rigorous self-check (no need to output the self-check content):
• Whether the questions fully comply with the task type and secondary task requirements;
• Whether the answers are accurate, verifiable, and free from speculation;
• Whether the questions contain no ambiguity, avoid repeated assessment points, and exhibit sufficient difficulty;
• Whether the domain-specific expression is correct (e.g., legal or classical-text domains);
• Whether the language of the questions and answers matches the long context.
If any criterion fails → You must return to Step 2 and regenerate the questions.

Input
`[Long Context]`: **{Long_Context}**
`[Primary Task]`: **{Primary_Task}**
`[Secondary Task]`: **{Secondary_Task}**

Output Constraints
1. Output three questions, and the question types must match the secondary task requirements.
2. The language of the questions, answers, and explanations must match the long text.
3. The output format must strictly contain: the questions, the answers, the question-design rationale, and the solution process. No additional content may be included.

C.2 Sample Verification Criteria

Task Objective
This task systematically and formally verifies question–answer samples initially generated by large models. The goal is to ensure that each sample meets the requirements **in task authenticity, reasoning correctness, and challenge level.**

Sample Generation Process
To balance sample authenticity with human labor costs, we adopt a human-model collaborative approach to generate test samples:
1. Multi-model Generation
For each long document, annotators generate three questions and answers using the following state-of-the-art models with specific prompts. Each model output includes: the question, the answer, the design rationale, and the solving process.
o Gemini-2.5-Pro
o GPT-5
o Claude-Sonnet-4
o DeepSeek-V3.2
o Qwen3-235B-A22B-Thinking-2507
2. Human Critique
Annotators critically review the model outputs and decide whether to adopt, modify, or discard each sample.
3. Secondary Review
Expert perform a final audit of generated samples; any sample failing the review returns for further revision.

Verification Tasks Overview
Annotators complete the following three assessments for each model-generated sample:
Task A: Question Compliance with Task Type and Context Requirement
Based on the model's "design rationale," annotators evaluate whether the question:
• Aligns with the task definition and context requirement.
• Relies on document content only, without using external information.
• Avoids subjective judgment or unverifiable inference.
• Remains on-topic and does not include unsupported facts.
Task B: Answer Correctness
Based on the model's "solving process," annotators evaluate whether:
• The reasoning steps are grounded in the document.
• The logic is internally consistent.
• The final answer is verifiable from the document.
• The answer does not contain hallucinations (fabricated information).
Task C: Question Challenge Level
Annotators input the question into the five models listed above and assess:
• At least one model answers incorrectly → the sample is considered challenging.
• If all five models answer correctly → the sample is not challenging.

Verification Procedure
Step 1: Read the model output, including the question, answer, design rationale, and solving process. Annotators fully understand the intended assessment points of the question.
Step 2: Execute Task A - evaluate question compliance:
o If the question does not match the task type or diverges from the document → discard the sample.
o If minor edits can fix the issue → modify and proceed to Step 3.
o If fully compliant → proceed to Step 3.
Step 3: Execute Task B - evaluate answer correctness by checking against the original document and reasoning process:
o If the answer is incorrect or the reasoning contains hallucinations/jumps → discard the sample.
o If minor edits can fix the issue → modify and proceed to Step 4.
o If fully correct → proceed to Step 4.
Step 4: Execute Task C - verify challenge level by testing the question with five models:
o If all models answer correctly → the question is not challenging; discard unless the question is highly valuable.
o If at least one model answers incorrectly → the question is challenging; proceed to Step 5.
Step 5: Select the best sample among all candidates:
o If multiple samples pass → select the optimal one (most aligned with definitions and highest difficulty) and add it to the sample set.

o If no sample passes → proceed to the next document.
Step 6: Submit for expert review:
o Annotators submit verified samples to long-context expert.
o If experts reject → revise according to expert feedback and resubmit until approval.

Precautions
Prohibited Actions
• Do not fabricate information not present in the document.
• Do not treat model hallucinations as facts.
• Do not use subjective questions.
• Do not generate unverifiable answers.
• Avoid overly simple questions.
Permitted/Encouraged Actions
• Retain questions that require deep reasoning, cross-paragraph inference, or implicit clues.
• Refine model-generated questions to improve quality.
• Use the model reasoning chain to assist verification, but final judgment must be human-led.

Sample Quality Checklist

Item	Check
Task type: Matches design	☐
Document reliance: Context only	☐
Clarity: Question is clear	☐
Answer verifiable: Checkable	☐
Reasoning: Grounded logic	☐
Challenge: At least one model fails	☐
Readability: Clear format	☐

C.3 Sample Rewriting Criteria

Task Objective
Standardize the prompts of different question types to a uniform format to improve the accuracy and success rate of subsequent automatic evaluation. Each question requires two prompts:
• **Non-Thinking Prompt** (does not output the thinking process)
• **Thinking Prompt** (outputs the thinking process first, then the answer)
Both types of prompts share the same structure, with only minor differences in Output Requirement and Output Example.

Unified Structure (Mandatory)
Each prompt follows the three sections below:
① ***First Section***: Task Description + Output Requirement (Mandatory)
• Task Description: Explain what the question asks to do.
• Output Requirement: Write differently for the Non-Thinking or Thinking version (see below).
② ***Second Section***: Supplementary Content (Optional)
If the question contains additional materials (e.g., abstract, full text, options, label list), include them in this section. Omit this section entirely if no supplementary content exists.
③ ***Third Section***: Output Example (Mandatory)
The output example must:
• Fully comply with the required answer format.
• Differ from the real answer content.
• Contain the "`[Answer]`" identifier.

Differences Between the Two Prompt Types (Core Part)
The following content is used to rewrite Output Requirement and Output Example. The rest (Task Description, Supplementary Content) remains unchanged.
1. Non-Thinking Prompt Specification
Output Requirement (Non-Thinking Version) — Fixed Template, Cannot Change
```
Output the "[Answer]" identifier first, and then output {elements} line by line, without any additional content.
```
Here, elements are provided by the annotator according to the question, e.g., sorted numbers, option letters, titles, years, or custom items. The format of each element must be clear, e.g., "Number Paragraph", "A", "Item1 Item2".
Output Example (Non-Thinking Version) — Fixed Format
```
Output example:
[Answer]
{Specific element 1}
{Specific element 2}
```
2. Thinking Prompt Specification
Output Requirement (Thinking Version) — Fixed Template, Cannot Change
```
Think step by step. After your thinking process,
```

output the "[Answer]" identifier, and then output {elements} line by line.
```
The elements are also specified according to the question.
*Output Example (Thinking Version) — Fixed Format*
```
Output example:
<Your thinking process>
[Answer]
{Specific element 1}
{Specific element 2}
```

**Prompt Rewriting Procedure (Mandatory for Annotators)**
The following steps are used to rewrite the original question into the standardized format.
***Step 1***: Break Down Original Question Content
Extract four parts from the original sample:
• Task Description
• Output Requirement (use fixed template for Non-Thinking / Thinking)
• Supplementary Content (if any)
• Output Example (write according to requirements)
***Step 2***: Construct Non-Thinking Prompt **Template**
```
Task Description Output the "[Answer]" identifier first, and then output {elements} line by line, without any additional content.

Supplementary Content (optional)

Output example:
[Answer]
{Specific element 1}
{Specific element 2}
```
***Step 3***: Construct Thinking Prompt **Template**
```
Task Description Think step by step. After your thinking process, output the "[Answer]" identifier, and then output {elements} line by line.

Supplementary Content (optional)

Output example:
<Your thinking process>
[Answer]
{Specific element 1}
{Specific element 2}
```
***Step 4***: Final Review
Check each item:
• The three-section structure is complete.
• Both Non-Thinking and Thinking formats use the fixed templates.
• Example format fully matches the answer format and differs from the real answer.
• "[Answer]" identifier exists and is correct.
• No mixing of Chinese and English punctuation.

### C.4 Answer Review Criteria

**Task Objective**
To ensure the accuracy and reliability of all sample answers, we perform consistent, standardized, and high-quality verification of long-context test samples. The verification process follows these principles:
• Human annotators are skilled at judging whether each component of an answer is correct → responsible for **precision**.
• Models can generate a diverse set of possible answer components → responsible for **recall**.
Therefore, during verification, we combine human judgment with model predictions to ensure data completeness.

**Verification Procedure**
***Step 1***: Generate model predictions For each sample, we pre-generate a list of predicted answers from five state-of-the-art models.
***Step 2***: Initial annotator verification (two annotators)
• *Step A*: Check the correctness of each component in the current answer (ensures precision).
• *Step B*: Check for any missing reasonable components, referring to model predictions if necessary (ensures recall).
***Step 3***: Result determination
• If both annotators identify no issues → mark the sample as problem-free and directly include it in the benchmark.
• If any annotator identifies a potential issue →
```

send the sample to a long-context expert for final judgment.
• The long-context expert determines, based on the issue type, whether the sample needs reconstruction.

Three Types of Potential Issues
During verification, three types of issues require identification:
① **Document Issues**
The provided document content is incomplete, lacking the necessary information to derive the current answer.
② **Question Issues**
The question description is unclear, incomplete, or ambiguous, preventing a reasonable derivation of the current answer.
③ **Answer Issues**
The document and question are fine, but the existing answer is incorrect or incomplete.

Annotator Operational Guidelines
1. Independent judgment
Both annotators must complete verification independently to avoid human consistency bias.
2. Ensure precision
Evaluate the correctness of each "component" in the answer to guarantee fine-grained assessment.
3. Ensure recall
Check for any missing components; the model prediction list serves as a reference.
o If a reasonable component appears in model predictions but is not covered in the current answer → mark as missing.
o If model predictions are unreasonable → do not consider it an issue.
4. Flag any doubts
Any uncertainty or potential error must be flagged for review by the long-context expert.

Long-Context Experts Guidelines
The long-context expert is responsible for:
1. Making final judgments on disputed samples.
2. Determining whether a sample requires reconstruction.
3. If reconstruction is needed, specifying the exact reason and corresponding step (modify document, rewrite question, or revise answer).

C.5 Sample Quality Evaluation Criteria

Task Objective
Each long-context question is scored along the following five evaluation dimensions. Each dimension allows scores of 0, 0.5, or 1.

Evaluation Dimensions and Criteria
1. Task Alignment
Definition: Measures whether the question aligns with the predefined task objectives, i.e., whether it matches the intended task design (25 secondary tasks in total).
Scoring Criteria:
• 1: Fully aligns with the task objective; the question format matches the task type closely.
• 0.5: Largely aligns, but the task intention is unclear or slightly deviates.
• 0: Clearly deviates from the task objective or uses an incorrect task type.
2. Context Requirement Alignment
Definition: Evaluates whether the question's dependence on document information matches the intended context requirement level. For example, Full questions require the entire document, while Partial questions require only local paragraphs.
Scoring Criteria:
• 1: The context dependency fully matches the intended level.
• 0.5: The dependency level is generally reasonable but slightly deviates.
• 0: The dependency level is incorrect (e.g., a Partial question requires the entire document).
3. Difficulty
Definition: Measures the challenge and discriminative power of the question, estimated based on the accuracy of five models answering the question.
Scoring Criteria:
• 1: Accuracy is between 0%–20%; at most one model answers correctly.
• 0.5: Accuracy is between 40%–60%; two to three models answer correctly.
• 0: Accuracy is between 80%–100%; four to five models answer correctly.
4. Authenticity
Definition: Measures whether the question reflects real user information needs and inquiry style, i.e., whether the question is natural and meaningful in practical scenarios.

Scoring Criteria:
• 1: Natural tone, fluent expression, reasonable question with authentic motivation.
• 0.5: Generally natural but slightly templated or artificial.
• 0: Clearly synthetic or inconsistent with real-world question style.
5. Answer Correctness
Definition: Measures whether the reference answer is accurate, consistent with the document, and reasonably verifiable.
Scoring Criteria:
• 1: Answer fully matches the document; facts are correct.
• 0.5: Answer is mostly correct with minor deviations or imprecise wording.
• 0: Answer is clearly incorrect or inconsistent with the document.

Scoring Procedure
1. **Independent Scoring**: Each sample is independently scored by three experts along the five dimensions.
2. **Score Levels**: Each dimension allows only 0, 0.5, or 1.
3. **Result Aggregation**: The final score for each dimension is the average of the three experts' scores.
4. **Notes**:
o Scoring strictly follows the criteria without subjective assumptions.
o For ambiguous cases, the context and task type may be considered, but consistency must be maintained.
o All scores require reasonable explanations for subsequent review.

D Models used for Difficulty Classification

We rank models by overall performance into three tiers (high/mid/low). Within each tier, we select five representative models that perform the best while covering diverse architectures. We classify the difficulty of the LongBench Pro samples based on these models.

- **High-performing models**: Gemini-2.5-Pro (Comanici et al., 2025), GPT-5 (OpenAI, 2025), Claude-4-Sonnet (Anthropic, 2025), DeepSeek-V3.2 (Liu et al., 2025), Qwen3-235B-A22B-Thinking-2507 (Yang et al., 2025)
- **Mid-performing models**: GLM-4.6 (, Z.ai), DeepSeek-V3-0324 (Liu et al., 2024), Kimi-K2-Instruct-0905 (Team et al., 2025b), Qwen3-30B-A3B-Instruct-2507 (Yang et al., 2025), MiniMax-M2 (Team, 2025)
- **Low-performing models**: Ministral-3-8B-Instruct-2512 (AI, 2025), Qwen3-8B (Yang et al., 2025), Qwen2.5-72B-Instruct (Qwen et al., 2025), Llama-3.1-405B-Instruct (AI, 2024), Gemma-3-27B-It (Team et al., 2025a)

E Data Statistics of LongBench Pro

We generate 5 samples evenly for 25 secondary tasks, 2 languages, and 6 lengths. In the end, LongBench Pro contains 1,500 samples, with the distributions of sample text types, languages, context requirements, lengths, and difficulty shown in Figure 7 (a). The distribution of samples across tasks is shown in Figure 7 (b).

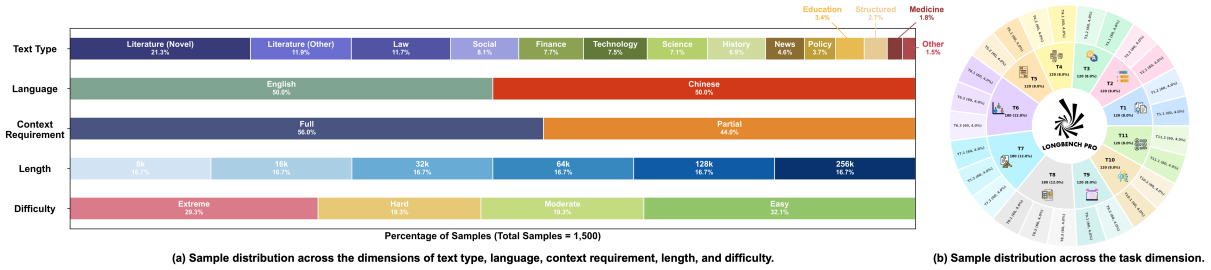
F Evaluation Settings

F.1 Evaluation Metrics

We use task-specific metrics summarized in Table 2. T1 (Retrieval & Ranking) is evaluated by NDCG@k. T2 and T6.3 (ordering-style tasks) use pairwise accuracy based on rank consistency. T3 and T11 are multiple-choice and use accuracy. For tasks with potentially multiple answer components extracted from the source text (T5, T6.2, T7, and T9), we use F1 to penalize spurious components. For tasks with a single canonical answer that is not directly copied from the source (T6.1, T8, and T10), we use SubEM. For T4 (Summary), we combine semantic similarity (SemSim) and ROUGE-L to balance semantic faithfulness and coverage. Each summarization sample includes three reference summaries. The metrics are first computed between the generated summary and each reference summary individually, and the maximum value for each metric is taken to reflect consistency with the best-matching reference. The final weighted score is calculated as:

$$\begin{aligned}\text{Score}_{\text{summary}} &= 0.5 \cdot \max_i \text{SemSim}(S_{\text{gen}}, S_{\text{ref}_i}) \\ &\quad + 0.5 \cdot \max_i \text{ROUGE-L}(S_{\text{gen}}, S_{\text{ref}_i})\end{aligned} \tag{2}$$

All metrics have a value range of $[0, 1]$. We report the average score over all samples and multiply by 100.



(a) Sample distribution across the dimensions of text type, language, context requirement, length, and difficulty.

(b) Sample distribution across the task dimension.

Figure 7: Overview of LongBench Pro sample distributions.

F.2 Evaluated Models

We evaluate 46 long-context models that vary in transparency (closed-source, such as GPT-5 (OpenAI, 2025); open-source, such as GPT-OSS-120B (OpenAI et al., 2025)), thinking mode (thinking, such as DeepSeek-R1 (DeepSeek-AI et al., 2025); mixed-thinking, such as DeepSeek-V3.2 (Liu et al., 2025); non-thinking, such as DeepSeek-V3-0324 (Liu et al., 2024)), size (3B, such as Ministral-3-3B-Instruct-2512 (AI, 2025); 1T, such as Kimi-K2-Instruct-0905 (Team et al., 2025b)), architecture (dense, such as Qwen3-32B (Yang et al., 2025); MoE, such as Qwen3-235B-A22B-Instruct-2507 (Yang et al., 2025)), and context length (128k, such as Gemma-3-27B-It (Team et al., 2025a); 1M, such as Gemini-2.5-Pro (Comanici et al., 2025)), with the goal of comprehensively assessing the long-context performance of current LLMs.

F.3 Inference Settings

We uniformly use each model's default inference parameters to run inference three times, and report both the general performance (the average score across multiple responses) and the upper-bound performance (Best-of-N, the highest score among the multiple responses; Pass@N, the probability that at least one response is completely correct). For models without default inference parameters, we set the temperature to 1.0. For thinking models, we use the non-thinking prompt and report their thinking scores. For mixed-thinking models, we use non-thinking prompts and report both the non-thinking and thinking scores under the disabled and enabled thinking states. For non-thinking (instruct) models, we use non-thinking prompts and thinking prompts separately to report the corresponding non-thinking and thinking scores. For the thinking score, models that support a 256k context length set the output length to 32k to enable more thorough reasoning, while for other models, we set the output length to 8k to reserve more budget for the input. For the non-thinking score, the output length is uniformly set to 1k. When the sample length exceeds the model's context length, we truncate the sample from the middle to an appropriate length for input, with the truncation length uniformly set to the model's context length minus the output length. Detailed inference parameter settings for different models are provided in Table 5, with the parameters taken from the open-source documentation of each model.

F.4 Truncation Length Setting

The actual output length of some thinking models is greater than the default 8k setting. For example, in MiniMax-M2, 538 out of 4,500 inferences have lengths far exceeding 8k. Setting the output length to 8k would result in many evaluation failures. Therefore, to reserve sufficient thinking space, we set the truncation length of DeepSeek-V3.2, GLM-4.6, and MiniMax-M2 to 120k, ensuring that the output length can be set to 32k.

In addition, there is a significant discrepancy between the effective context length and the claimed context length for some models. Table 4 presents a comparison of GLM-4.6's non-thinking scores at truncation lengths of 190k and 120k. When the truncation length is set to 190k, for 256k samples (whose length approaches 190k), the model outputs become unstable, leading to a sharp drop in metrics. This indicates that GLM-4.6's effective context length significantly deviates from its claimed context length. This is also one of the reasons why we set its truncation length to 120k.

Truncation Length	Sample Length					
	8k	16k	32k	64k	128k	256k
190k	53.74	50.76	51.93	45.60	37.73	**2.55**
120k	53.98	49.88	52.26	46.18	38.68	34.14

Table 4: Non-thinking scores of GLM-4.6 on samples of different lengths under varying truncation lengths.

Model	Model Type	Context Length Input	Context Length Output	Truncation Length	Output Length Non-Thk.	Output Length Thk.	Temperature Non-Thk.	Temperature Thk.
Gemini-2.5-Pro	Thinking	1M	64k	1M	-	32k	-	1.0
Gemini-2.5-Flash	Mixed	1M	64k	1M	1k	32k	1.0	1.0
Gemma-3-27B-It	Instruct	128k		120k	1k	8k	1.0	1.0
Gemma-3-12B-It	Instruct	128k		120k	1k	8k	1.0	1.0
Gemma-3-4B-It	Instruct	128k		120k	1k	8k	1.0	1.0
GPT-5	Thinking	272k	128k	272k	-	32k	-	1.0
GPT-4o	Instruct	128k		120k	1k	8k	1.0	1.0
GPT-OSS-120B	Thinking	128k		120k	-	8k	-	1.0
GPT-OSS-20B	Thinking	128k		120k	-	8k	-	1.0
Claude-4-Sonnet	Mixed	1M	64k	1M	1k	32k	1.0	1.0
Claude-3.7-Sonnet	Mixed	200k	128k	200k	1k	32k	1.0	1.0
DeepSeek-V3.2 *	Mixed	160k		120k	1k	32k	1.0	1.0
DeepSeek-V3.1	Mixed	128k		120k	1k	8k	0.6	0.6
DeepSeek-R1-0528	Thinking	128k		120k	-	8k	-	0.6
DeepSeek-R1	Thinking	128k		120k	-	8k	-	0.6
DeepSeek-V3-0324	Instruct	128k		120k	1k	8k	0.3	0.3
Qwen3-235B-A22B-Thinking-2507	Thinking	256k		224k	-	32k	-	0.6
Qwen3-235B-A22B-Instruct-2507	Instruct	256k		224k	1k	32k	0.7	0.7
Qwen3-Next-80B-A3B-Thinking	Thinking	256k		224k	-	32k	-	0.6
Qwen3-Next-80B-A3B-Instruct	Instruct	256k		224k	1k	32k	0.7	0.7
Qwen3-30B-A3B-Thinking-2507	Thinking	256k		224k	-	32k	-	0.6
Qwen3-30B-A3B-Instruct-2507	Instruct	256k		224k	1k	32k	0.7	0.7
Qwen3-4B-Thinking-2507	Thinking	256k		224k	-	32k	-	0.6
Qwen3-4B-Instruct-2507	Instruct	256k		224k	1k	32k	0.7	0.7
Qwen3-32B	Mixed	128k		120k	1k	8k	0.7	0.6
Qwen3-14B	Mixed	128k		120k	1k	8k	0.7	0.6
Qwen3-8B	Mixed	128k		120k	1k	8k	0.7	0.6
Qwen3-4B	Mixed	128k		120k	1k	8k	0.7	0.6
Qwen2.5-72B-Instruct	Instruct	128k		120k	1k	8k	0.7	0.7
GLM-4.6 *	Mixed	198k		120k	1k	32k	1.0	1.0
GLM-4.5	Mixed	128k		120k	1k	8k	1.0	1.0
Kimi-K2-Instruct-0905	Instruct	256k		224k	1k	32k	0.6	0.6
MiniMax-M2 *	Thinking	192k		120k	-	32k	-	1.0
MiniMax-Text-01	Instruct	4M		1M	1k	32k	1.0	1.0
Ministral-3-14B-Instruct-2512	Instruct	256k		224k	1k	32k	0.1	0.1
Ministral-3-8B-Instruct-2512	Instruct	256k		224k	1k	32k	0.1	0.1
Ministral-3-3B-Instruct-2512	Instruct	256k		224k	1k	32k	0.1	0.1
Magistral-Small-2509	Thinking	128k		120k	-	8k	-	0.7
Mistral-Small-3.2-24B-Instruct-2506	Instruct	128k		120k	1k	8k	0.15	0.15
Mistral-Large-Instruct-2411	Instruct	128k		120k	1k	8k	1.0	1.0
Ministral-8B-Instruct-2410	Instruct	128k		120k	1k	8k	1.0	1.0
Llama-3.1-405B-Instruct	Instruct	128k		120k	1k	8k	0.6	0.6
Llama-3.3-70B-Instruct	Instruct	128k		120k	1k	8k	0.6	0.6
Llama-3.1-70B-Instruct	Instruct	128k		120k	1k	8k	0.6	0.6
Llama-3.1-8B-Instruct	Instruct	128k		120k	1k	8k	0.6	0.6
Llama-3.2-3B-Instruct	Instruct	128k		120k	1k	8k	0.6	0.6

Table 5: Detailed inference parameter settings. **Non-Thk.** denotes Non-Thinking, and **Thk.** denotes Thinking. **Length** is uniformly measured by the number of tokens. *: Although these models support longer context lengths, we set their truncation length uniformly to 120k and the thinking output length to 32k to enable more thorough thinking. Appendix F.4 provides specific notes regarding this part.

G Additional Analysis

G.1 Performance across different tasks.

Figure 8 shows mainstream LLMs' performance across different tasks, corresponding to detailed analysis in Section 5.3.

G.2 Upper-Bound Performance

To filter out metric deviations caused by generation instability, we report the trend of Best-of-N metrics for mainstream LLMs, as shown in Figure 9. All models show a clear monotonic increase. Gemini-2.5-Pro and GPT-5 exhibit strong stability, as their

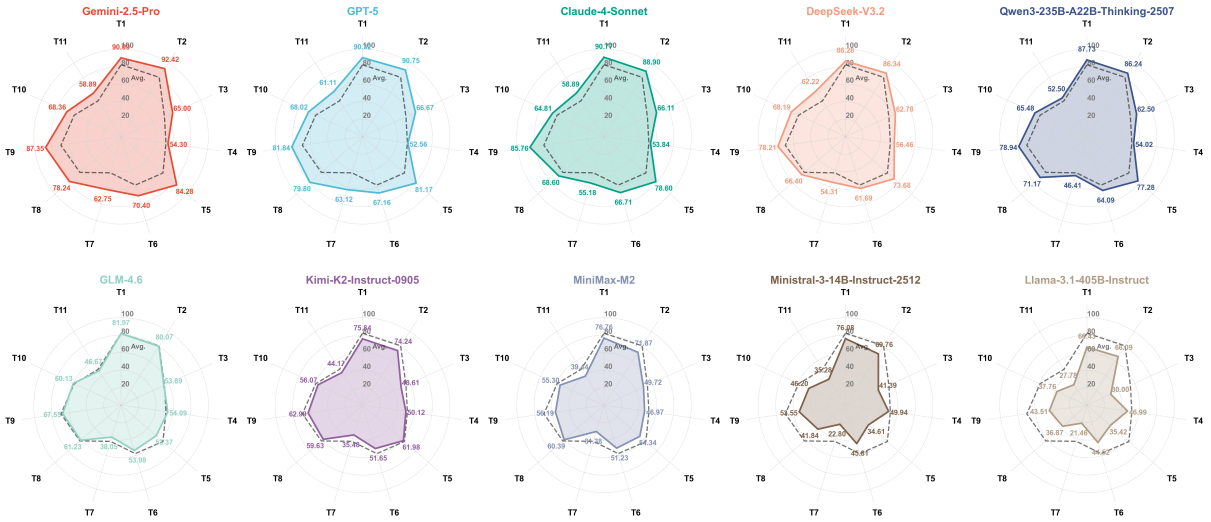


Figure 8: Performance across different tasks.

single-shot performance is already high and the marginal gains from additional inference converge quickly. In contrast, models such as Qwen3-235B-A22B-Thinking-2507 exhibit markedly high potential: increasing N significantly corrects the bias in their initial reasoning, enabling them to make a leap toward the top tier.

LongBench Pro adopts a fine-grained scoring mechanism, granting partial credit for partially correct answers. This mechanism enables a more nuanced differentiation of performance across models; however, it fails to fully reflect the intrinsic difficulty of LongBench Pro. Therefore, we further report the trend of Pass@N metrics for mainstream LLMs on LongBench Pro, as shown in Figure 10.

The above upper-bound evaluation substantiates the effectiveness and soundness of LongBench Pro. Under N = 3, LongBench Pro still maintains the following properties:

(1) Discriminability: There remains a clear performance gap across model tiers, indicating that LongBench Pro assesses deep long-context understanding rather than surface-level tricks that can be compensated for by probabilistic guessing.

(2) Difficulty: Even the strongest model, Gemini-2.5-Pro, achieves only a Pass@3 of 10.68 on the Extreme samples. After excluding factors related to model instability, the benchmark still exhibits a substantial headroom, sufficient to support the evaluation of more capable models.

G.3 Comparison Across Construction Strategies

We uniformly sample 50 documents based on secondary task categories, lengths, and languages, and construct samples using three different strategies to evaluate the effectiveness of our strategy. The compared sample-construction strategies include human-only, model-only, and human-model collaboration (ours). Figure 5 (a) presents the time cost of constructing samples of different lengths under these strategies. The time required by the human-only strategy increases exponentially with sample length, while the model-only strategy remains consistently low. The human-model collaborative strategy falls between the other two and increases slowly with sample length, highlighting the substantial efficiency gains enabled by model involvement in the sample-construction process.

We also evaluate the quality of samples constructed using three strategies. Specifically, we design a sample quality assessment framework comprising five dimensions. According to this framework, each sample is scored by three experts on three levels, with scores ranging from 0 to 1. The results show that samples constructed via the human-model collaborative strategy achieve the highest average quality score (0.9609 ± 0.0415), outperforming those constructed by the human-only strategy (0.9484 ± 0.0450) and the model-only strategy (0.8964 ± 0.0536). The Fleiss' Kappa among the three experts is 0.76, indicating high agreement. Figure 5 (b) shows the quality of samples constructed by different strategies across various dimensions. Thanks to our clear and explicit def-

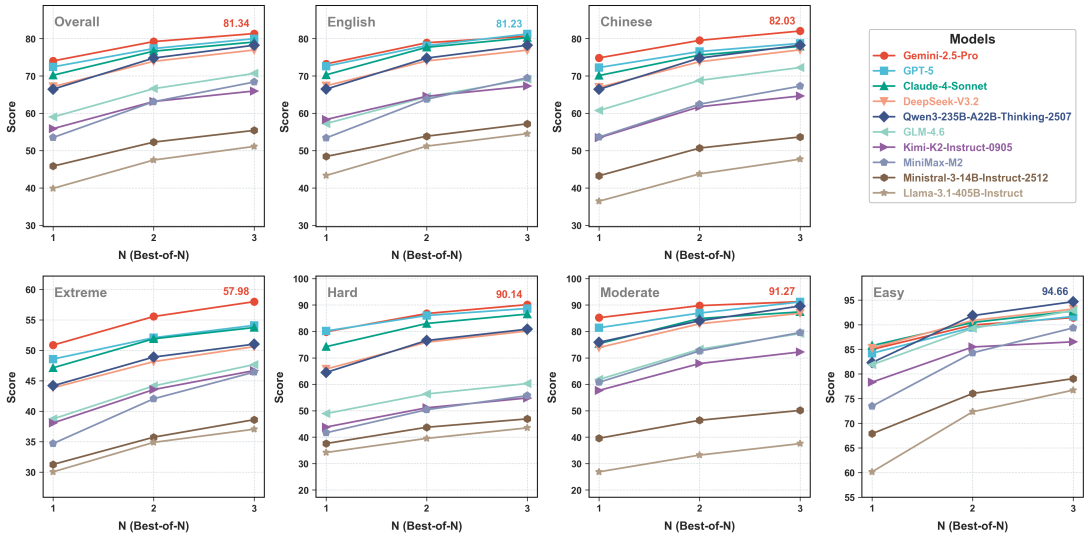


Figure 9: Trends in Best-of-N metrics.

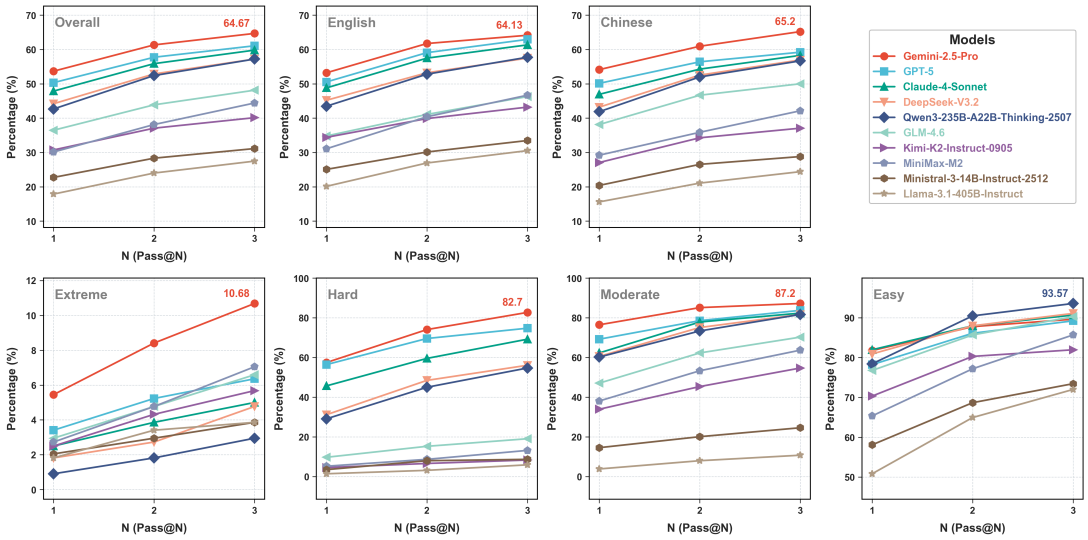


Figure 10: Trends in Pass@N metrics.

initions of task and context requirements, as well as fully authentic natural text, all three sample construction strategies achieve consistently high scores in task alignment, context requirement alignment, and authenticity. On the difficulty dimension, the model-only strategy exhibits a relatively low level due to the absence of human sample filtering. In terms of answer correctness, the human-only strategy, without model assistance, tends to has omissions in answer components, resulting in the lowest correctness. The model-only strategy, without human error correction, tends to produces erroneous hallucinated answers, leading to slightly lower correctness. The human-model collaboration strategy effectively compensates for the weaknesses of both strategies, achieving the highest answer correctness.

H Annotator Statistics and Compensation

The construction of LongBench Pro involves a total of 63 annotators, with the statistical distribution shown in Figure 11. The 63 annotators are divided into 51 general annotators and 12 long-context experts (all long-text experts have at least one year of annotation experience and receive a two-month specialized training in long-context annotation). The ages of the annotators mainly range from 23 to 32 years, the gender ratio is balanced, and their major backgrounds are diverse. In addition, more than half of the annotators have over one year of annotation experience. Most annotators hold a bachelor's degree, and approximately 25% possess a graduate degree. Annotators are compensated at a rate of 50 RMB per hour.

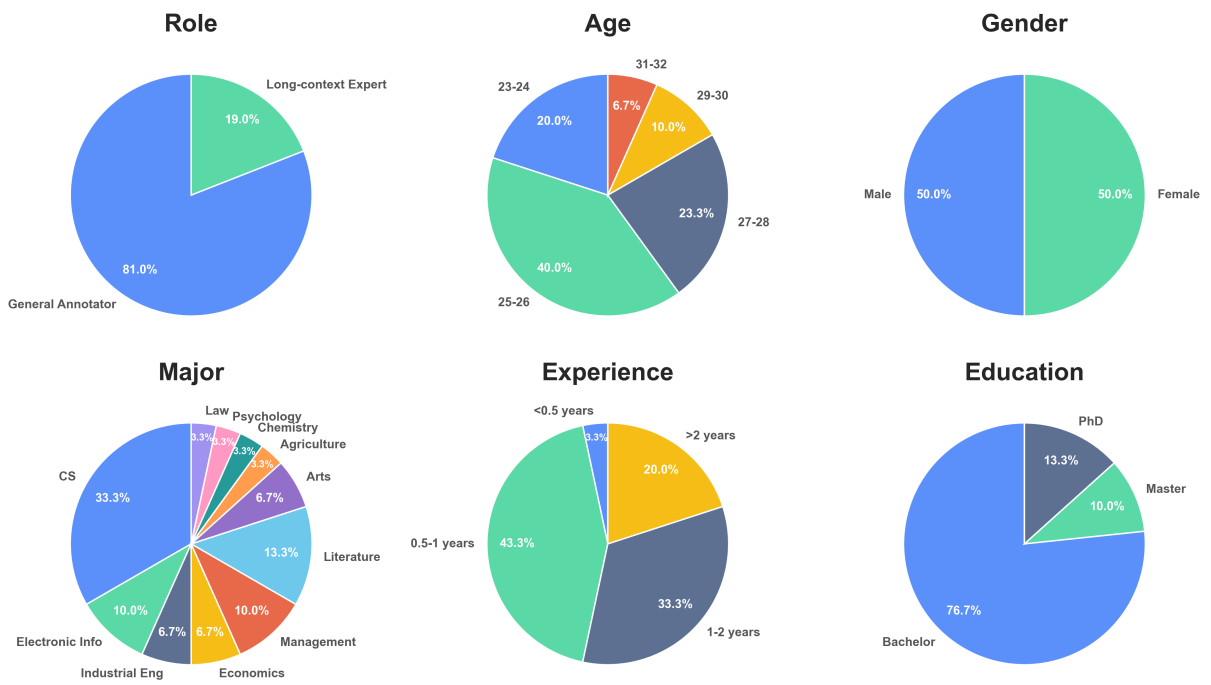


Figure 11: Distribution of annotators' role, age, gender, major, experience, and education.

I Related Works

Long-context evaluation measures whether LLMs can reliably retrieve, integrate, and reason over evidence that is sparse and distributed across lengthy documents, and it must address confounders such as positional effects and the gap between *advertised* context length and *effective* reasoning length (Liu et al., 2023). Existing benchmarks span a spectrum from controlled probes to realistic, human-verified tasks: synthetic benchmarks such as RULER (Hsieh et al., 2024), MRCR (OpenAI, 2025), and GSM-∞ (Zhou et al., 2025) provide scalable diagnostics of usable context size, while long-document NLP suites (e.g., SCROLLS/ZeroSCROLLS (Shaham et al., 2022, 2023)) and standardized evaluation protocols (e.g., L-Eval and its length-adaptable extension Ada-L-Eval (An et al., 2023; Wang et al., 2024)) broaden task coverage and comparability. More realistic benchmark datasets further emphasize natural documents and deeper reasoning, including LongBench (Bai et al., 2023) and LongBench v2 (Bai et al., 2025), as well as mixed synthetic/natural stress tests such as ∞Bench (Zhang et al., 2024) and multilingual evaluation such as CLongEval (Qiu et al., 2024); methodology-focused work like HELMET (Yen et al., 2024) argues for systematic designs and analyses to avoid over-optimistic conclusions. In contrast to prior benchmarks that are primarily synthetic probes, protocol suites, or narrower in language/task coverage, LongBench Pro is built on fully natural long documents with bilingual (EN/ZH) coverage and diverse tasks/metrics, and it supports fine-grained analysis via multi-dimensional categorization (context requirement, length, difficulty) enabled by a scalable human-model collaborative construction pipeline.