Constructing Thunder Korean Benchmark Suite for Reliable Evaluation of Foundation Models

Yeonkyoung So [1] **Jongmin Kim** [1] **Sungmok Jung** [1] **Gyuseong Lee** [1] **Sangho Kim** [1] **Jongyeon Park** [1] **Joonhak Lee** [1] **Seho Pyo** [1] **Gyeongje Cho** [1] **Seorin Kim** [1] **JiSoo Kim** [1] **Suyoung Park** [1] **Hyunji Park** [1] **Yelim Ahn** [1] **Yeongho Seo** [2] **Jaejin Lee** [1 2]

Abstract

Reliable evaluation of foundation models in Korean requires benchmarks that measure intended capabilities rather than artifacts introduced by translation, localization, or evaluation protocol. In practice, Korean evaluation often adapts established English benchmarks, but literal translation can alter task difficulty, reduce prompt naturalness, or change what the task is intended to evaluate. We present a Thunder Korean Benchmark Suite comprising Ko-ARC, Ko-GSM8K, Ko-EQ-Bench, Ko-WinoGrande, Ko-LAMBADA, and Ko-IFEval, covering six capabilities across 9,396 items. Rather than treating translation as a single preprocessing step, we construct each subset using one of three routes: expert-reviewed translation and localization, direct Korean construction, or a hybrid of localized adaptation and Korean-specific redesign. For multiple-choice subsets, we also report NPSQ-based accuracy to assess whether models rely on question evidence rather than superficial choice preference. Results show that Korean benchmark scores depend on task construction, scoring rules, prompting, and model post-training. We further find that different scoring methods can lead to different interpretations depending on the task, highlighting the need to report benchmark scores together with their evaluation protocol.

1. Introduction

Large language models (LLMs) have rapidly advanced in recent years and are now used across a wide range of domains requiring language understanding, generation, and reasoning (Brown et al., 2020; Sindhu et al., 2024; Zhou et al., 2025). As these models become more capable and more widely deployed, it becomes increasingly important to evaluate them using objective and standardized criteria (Chang et al., 2024; Desai et al., 2024). Benchmarks play a central role in this process by providing shared tasks for comparing models and tracking progress across capabilities such as knowledge, reasoning, contextual understanding, and instruction following (Clark et al., 2018; Sakaguchi et al., 2021; Cobbe et al., 2021; Hendrycks et al., 2021; Liang et al., 2022; Paech, 2023; Zhou et al., 2023; Wang et al., 2024b). However, benchmark scores are only useful when the benchmark measures the intended capability rather than artifacts of dataset construction or evaluation (Durmus et al., 2022; Wu et al., 2022; Bean et al., 2026).

This issue is particularly important for languages outside a small set of major evaluation languages. English benchmarks have become broad and diverse enough that researchers can evaluate models on many capabilities using widely recognized test sets (Srivastava et al., 2023; Shen et al., 2024). For Korean, the situation has improved, and several valuable benchmarks have been introduced (Park et al., 2021; Son et al., 2024; 2025b). Still, it remains difficult to find Korean benchmark suites that both cover a broad range of capabilities and support close comparison with widely used English benchmarks.

For this reason, Korean model evaluation often depends on adapting established English benchmarks. The most straightforward way to do this is direct translation, but translation-only evaluation can distort the benchmark itself. Literal translation, mistranslation, stylistic inconsistency, and cultural mismatch may change task difficulty, alter the naturalness of the prompt, or even shift what the task is testing (Xu et al., 2025; Chua et al., 2025; Doddapaneni et al., 2025). In such cases, differences in benchmark scores become harder to interpret, since they may reflect properties of the translation as much as the model's actual capability in Korean.

In this work, we present a Thunder Korean Benchmark Suite for evaluating foundation models across multiple capabil-

[1]Graduate School of Data Science, Seoul National University [2]Dept. of Computer Science, Seoul National University. Correspondence to: Jaejin Lee <jaejin@snu.ac.kr>.

Accepted to the 1st Workshop on Combining Theory and Benchmarks, CTB@ICML 2026, Seoul, South Korea. Copyright 2026 by the author(s).

ities, including science question answering, mathematical reasoning, dialogue sentiment analysis, contextual reasoning, literary context understanding, and instruction following. Our construction pipeline combines machine translation with expert review, linguistic correction, cultural localization, and cross-validation. For tasks that cannot be transferred appropriately by translation alone, we redesign the benchmark to preserve the intended capability in Korean rather than treating translation as a simple preprocessing step.

Using the resulting benchmark suite, we evaluate both Korean LLMs and global LLMs with multilingual capability and examine how model behavior differs across tasks. For multiple-choice subsets, we also examine whether standard accuracy-based evaluation may be affected by superficial properties of the answer choices, which can make benchmark results harder to interpret (Cho et al., 2026). More broadly, our study suggests that benchmarking in languages beyond English requires careful consideration of both how benchmarks are transferred across languages and how evaluation protocols shape the interpretation of benchmark scores. The evaluation code and benchmark resources are publicly available.[1]

2. Related Work

2.1. Benchmarking Foundation Models

Benchmarks have become a standard tool for evaluating foundation models. By providing shared tasks and standardized metrics, they enable researchers to compare models across capabilities such as factual knowledge, reasoning, language understanding, code generation, and instruction following. Widely used examples include MMLU (Hendrycks et al., 2021) for multi-domain knowledge, ARC (Clark et al., 2018) for science question answering, GSM8K (Cobbe et al., 2021) for mathematical reasoning, WinoGrande (Sakaguchi et al., 2021) and HellaSwag (Zellers et al., 2019) for commonsense and contextual reasoning, HumanEval (Chen et al., 2021) for code generation, and IFEval (Zhou et al., 2023) for instruction following.

Beyond individual task-specific datasets, holistic evaluation frameworks have attempted to aggregate diverse tasks into broader evaluation suites. HELM (Liang et al., 2022) evaluates models across multiple scenarios and metrics, while BIG-Bench (Srivastava et al., 2023) collects a large set of tasks designed to probe a wide range of model behaviors. As model performance has improved, more challenging benchmarks have also been introduced to better differentiate frontier models. Examples include MMLU-Pro (Wang et al., 2024b), GPQA (Rein et al., 2024), and expert-level academic benchmarks (Phan et al., 2025). Domain-specific benchmarks, such as LegalBench (Guha et al., 2023) and MedQA (Jin et al., 2021), further show that general-purpose benchmarks are often insufficient for evaluating specialized capabilities.

2.2. Evaluation Sensitivity

Benchmark scores are not always straightforward indicators of model capability. Recent work has shown that evaluation outcomes can depend on factors that are not directly related to the target skill being tested. For example, small changes in prompt wording or formatting can lead to different measured performance, suggesting that benchmark results may depend on the particular evaluation template used (Sclar et al., 2023; Polo et al., 2024). In multiple-choice question answering, model predictions can also be sensitive to the order, wording, length, format, or scoring of answer choices (Pezeshkpour & Hruschka, 2024; Zheng et al., 2023; Wang et al., 2024a; Molfese et al., 2025). More fundamentally, models may sometimes rely on the answer choices themselves, rather than on the question, so high accuracy does not always imply that the model solved the task in the intended way (Balepur et al., 2024; 2025; Wang et al., 2025; Cho et al., 2026).

These findings suggest that benchmark design involves more than selecting a task and reporting accuracy. The dataset construction process, prompt format, scoring rule, and evaluation protocol can all affect how benchmark scores should be interpreted. We account for these concerns in our multiple-choice evaluation by reporting NPSQ-based accuracy, which helps distinguish question-driven evidence from answer-choice artifacts.

2.3. Korean LLM Evaluation

Korean LLM evaluation has developed rapidly in recent years, but remains distributed across different goals, including general language understanding, cultural knowledge, professional knowledge, instruction following, and reasoning. Early resources such as KLUE, KorNLI/KorSTS, and KoBEST have provided important foundations for Korean NLU evaluation (Park et al., 2021; Ham et al., 2020; Jang et al., 2022). KLUE has introduced a broad Korean language understanding benchmark, while KorNLI and KorSTS have adapted English NLI and STS resources into Korean through translation and human post-editing. KoBEST has further introduced Korean adaptations of several established tasks, motivated by the limitation that machine-translated benchmarks may not fully reflect Korean linguistic characteristics.

More recent benchmarks have expanded Korean evaluation toward native Korean sources and Korean-specific knowl-

[1] Code: https://github.com/mcrl/korean_benchmarks. Benchmark resources: https://huggingface.co/collections/thunder-research-group/snu-thunder-llm-korean-benchmark-suite.

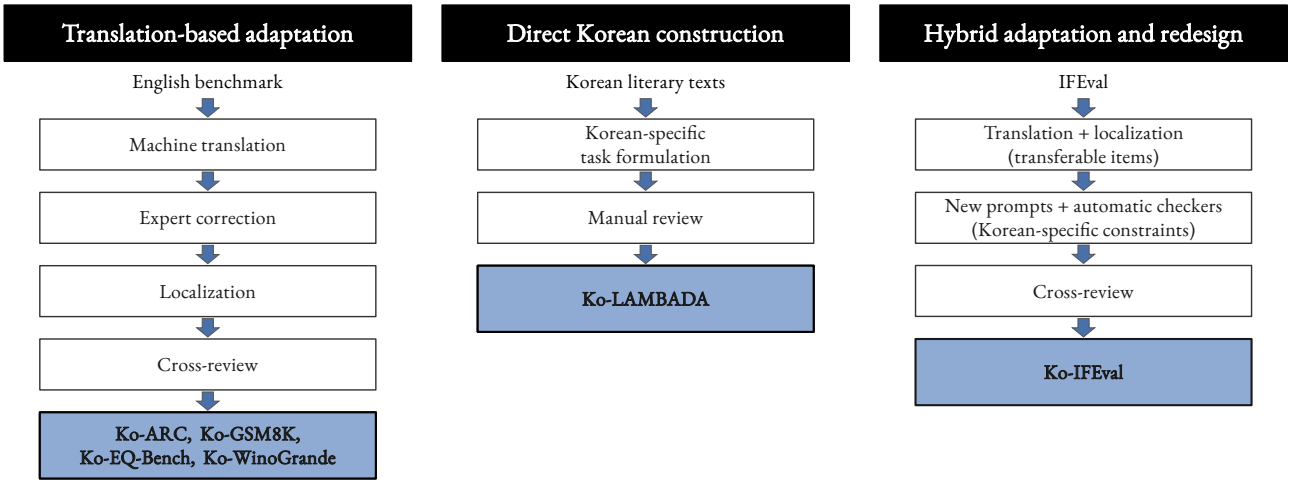

Figure 1. Construction routes of the Thunder Korean Benchmark Suite. Translation-based tasks are adapted from English benchmarks through machine translation, expert correction, localization, and cross-review. Directly constructed tasks are built from Korean sources, while hybrid tasks combine translated items with Korean-specific prompts and automatic checkers.

edge. HAE-RAE Bench evaluates Korean cultural and contextual knowledge (Son et al., 2024), while CLIcK draws from official Korean exams and textbooks to assess cultural and linguistic intelligence (Kim et al., 2024). KMMLU provides large-scale expert-level multiple-choice questions collected from Korean exams (Son et al., 2025b), with later variants such as KMMLU-Redux and KMMLU-Pro focusing on question quality and professional knowledge (Hong et al., 2025). Other recent benchmarks target more specific dimensions, including national alignment and social values, Korean commonsense, and Korean mathematical reasoning (Lee et al., 2024; Seo et al., 2024; Son et al., 2025a; Ko et al., 2025). In addition, the Open Ko-LLM Leaderboard series has attempted to provide shared evaluation frameworks for Korean LLMs, but leaderboard-based efforts remain limited when the underlying datasets, evaluation details, or long-term maintenance are not fully available (Park et al., 2024; Kim et al., 2025).

These efforts show that Korean LLM evaluation is no longer limited to a small number of datasets. They also show a clear shift from simply machine-translating English resources toward constructing tasks that better reflect Korean language use, culture, and real-world evaluation needs. Still, this expanding landscape raises a design question that is central to our work: how to determine when an English benchmark should be translated for comparability, localized for Korean naturalness, or redesigned because the original formulation no longer measures the same capability in Korean.

Our work addresses this question by constructing a multi-task Korean benchmark suite that combines expert-reviewed translation, localization, and task redesign. We keep links to established English benchmarks where comparison is useful, but modify or reconstruct tasks when direct transfer would distort the intended evaluation. Together with the evaluation sensitivity considerations discussed above, this construction strategy supports more interpretable Korean benchmark results.

3. Constructing Thunder Korean Benchmarks

3.1. Overview of the Thunder Korean Benchmark Suite

The Thunder Korean Benchmark Suite evaluates foundation models across six capabilities: science question answering, mathematical reasoning, dialogue sentiment analysis, commonsense and contextual reasoning, long-context literary understanding, and instruction following. It is built around established English benchmarks—ARC (Clark et al., 2018), GSM8K (Cobbe et al., 2021), EQ-Bench (Paech, 2023), WinoGrande (Sakaguchi et al., 2021), LAMBADA (Paperno et al., 2016), and IFEval (Zhou et al., 2023)—while adapting each task to Korean according to its linguistic and cultural characteristics.

We construct the suite using three routes, as illustrated in Figure 1. First, Ko-ARC, Ko-GSM8K, Ko-EQ-Bench, and Ko-WinoGrande are adapted from their English counterparts through translation, expert correction, localization, and cross-review. Second, Ko-LAMBADA is directly constructed from Korean literary texts because the original English task formulation does not transfer cleanly to Korean. Third, Ko-IFEval follows a hybrid strategy: transferable instructions are translated and localized, while Korean-specific constraints and automatic checkers are newly developed.

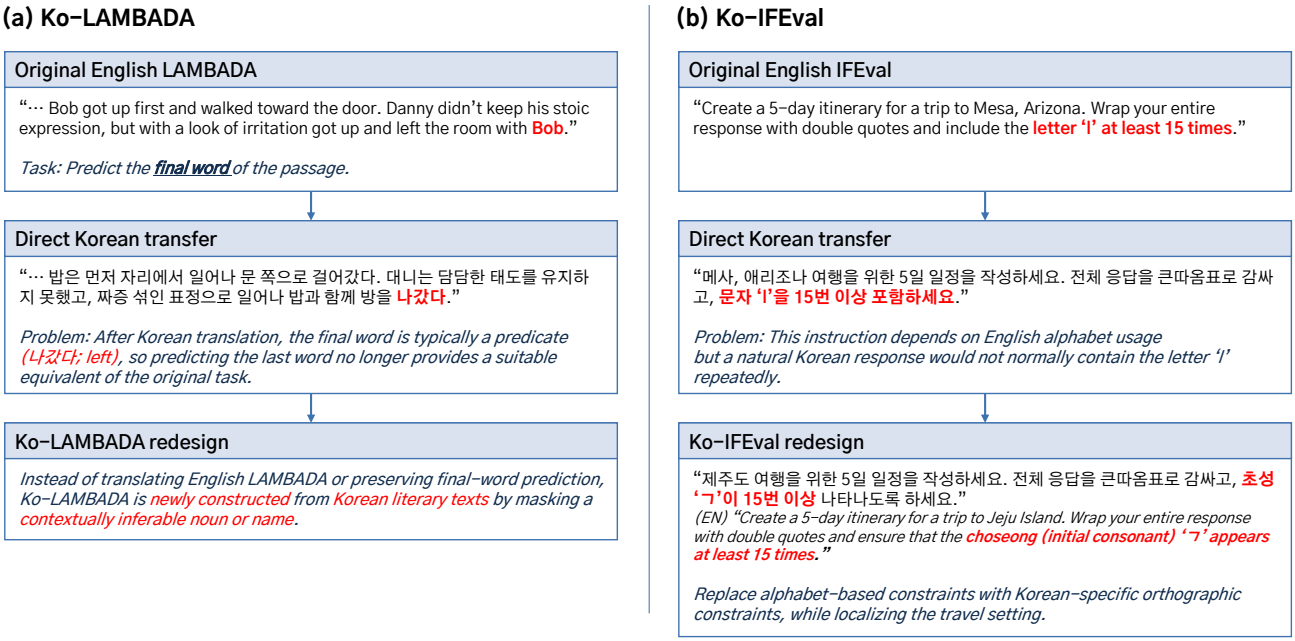

Figure 2. Examples illustrating why task redesign is needed when adapting English benchmarks to Korean. Each panel contrasts direct transfer with the corresponding redesigned Korean task formulation.

This task-level construction strategy allows us to retain comparability with established English benchmarks where appropriate, while redesigning tasks when direct transfer would change the capability being evaluated. The following subsections describe how English benchmarks are adapted to Korean, which tasks require redesign, and the resulting dataset statistics.

3.2. Adapting English Benchmarks to Korean

For Ko-ARC, Ko-GSM8K, Ko-EQ-Bench, and Ko-WinoGrande, we adapt the corresponding English benchmarks through a four-stage pipeline: initial machine translation, expert correction, localization, and independent cross-review. The goal is to preserve the original task structure and answer key where possible, while removing artifacts that would make the Korean version unnatural or misleading. Additional dataset-specific correction and localization rules are provided in Appendix A.

We first translate the English source items using the DeepL API[2]. We treat the machine-translated output only as a draft, since direct translation often produces literal phrasing, inconsistent terminology, and unnatural Korean expressions. Native Korean authors then revise each item, correcting both translation artifacts and issues in the original English items, such as typos, duplicate questions, incorrect answer labels, and incorrectly formatted solution annotations. For domain-specific tasks, authors also verify that scientific and mathematical terminology follows standard Korean usage and that the translated quantities, conditions, and explanations remain consistent with the answer.

Localization is applied when a literal translation would make the item culturally unnatural or change the intended difficulty. We adjust names, units, currency, everyday objects, and culturally specific scenarios while preserving the causal or logical structure of the original item. For example, in math problems, we convert units and currencies to Korean conventions while keeping the arithmetic simple. We also replace culturally unfamiliar situations with Korean equivalents when the answer remains unchanged. For science questions, localization is applied selectively. We standardize Korean terminology and wording, but we do not replace foreign scientific examples simply to make them Korean when the original context is part of the scientific content.

Finally, each item is independently cross-reviewed by an author who did not participate in the earlier correction and localization stage. This pass checks whether the Korean item is fluent, whether the answer key remains valid, and whether localization has introduced unintended ambiguity. If translation or localization makes the question ambiguous or changes the correct answer, we revise or remove the item. This final check reduces translation artifacts while keeping the Korean dataset aligned with the original benchmark.

[2] https://www.deepl.com/products/api

Table 1. Composition of the Thunder Korean Benchmark Suite. Construction type indicates whether the benchmark is translated and refined from an English source, newly constructed from Korean sources, or combines translated items with newly constructed items for Korean-specific constraints. Ko-IFEval consists of 541 translated items and 300 newly constructed items.

Benchmark	Capability	Items	Construction	English Source
Ko-ARC (easy)	Science QA	2,376	Translated + Refined	ARC-Easy
Ko-ARC (challenge)	Science QA	1,167	Translated + Refined	ARC-Challenge
Ko-GSM8K	Mathematical reasoning	1,319	Translated + Refined	GSM8K
Ko-EQ-Bench	Dialogue-based emotional inference	171	Translated + Refined	EQ-Bench
Ko-WinoGrande	Commonsense/contextual reasoning	1,267	Translated + Refined	WinoGrande
Ko-LAMBADA	Long-context literary understanding	2,255	Newly Constructed	LAMBADA
Ko-IFEval	Instruction following	841	Translated + Newly Constructed	IFEval
Total		**9,396**		

3.3. Task Redesign for Korean

The adaptation pipeline above is appropriate when an English task can be expressed in Korean without changing the capability being evaluated. For Ko-LAMBADA and Ko-IFEval, however, direct transfer is insufficient because the original task formulation depends partly on English-specific linguistic or formatting assumptions. We therefore redesign these tasks for Korean rather than treating translation as the default. Figure 2 provides representative examples of these two redesign cases.

Ko-LAMBADA. Ko-LAMBADA is constructed directly from copyright-cleared Korean literary texts, primarily collected from Gongu Madang (공유마당)[3], instead of translating the original English LAMBADA. We choose direct construction for two reasons. First, the original LAMBADA task asks models to predict the final word of an English passage. In many cases, this final word is a noun or name that must be inferred from the preceding context. In Korean, however, sentences typically end with predicates, so final-word prediction would often test verb endings rather than long-context understanding. Second, machine-translated literary passages often sound unnatural in Korean and tend to flatten literary expressions. We therefore construct new items from Korean literary sources rather than translating English literary passages.

To build candidate items, we automatically extract passage spans of an appropriate length from the collected Korean texts. We select spans that contain a target noun or name that can be masked, together with another in-context noun or name that can serve as a plausible distractor. The target is then masked in the passage, and candidate items are manually reviewed to ensure that the answer is inferable from context, that the distractor is plausible but not equally valid, and that the masked sentence remains grammatical and natural.

[3] https://gongu.copyright.or.kr/gongu/main/main.do

Ko-IFEval. Ko-IFEval follows a hybrid strategy. Some original IFEval instructions transfer naturally to Korean after translation and localization, such as constraints involving required keywords, forbidden words, or output format. Other constraints, on the other hand, depend on English-specific writing conventions, such as alphabet-based character constraints or word-count rules that do not map cleanly onto Korean morphology and spacing. For these cases, we add Korean-specific instruction types and adapt the automatic checkers accordingly. Examples include constraints based on Korean choseong (초성; leading consonant), Korean character-count rules, and formatting conventions that can be checked automatically for Korean outputs. As a result, Ko-IFEval keeps the original goal of evaluating instruction following with automatically checkable constraints, while extending the constraint set to Korean.

3.4. Dataset Statistics

Table 1 summarizes the resulting benchmark suite. The suite contains 9,396 evaluation items across six capabilities. The largest subset is Ko-ARC, with 3,543 science question-answering items across easy and challenge subsets, followed by Ko-LAMBADA with 2,255 literary context-understanding items. Ko-GSM8K, Ko-WinoGrande, Ko-IFEval, and Ko-EQ-Bench contribute 1,319, 1,267, 841, and 171 items, respectively.

4. Experiments

4.1. Evaluation Setup

Models. We evaluate both open-weight models and closed API models. Closed API models are evaluated through their official APIs: Claude Haiku 4.5 (`claude-haiku-4-5-20251001`) (ANTHROPIC, 2025) and Claude Opus 4.7 (`claude-opus-4-7`) (**?**) through the Anthropic API[4], GPT-5-mini

[4] https://platform.claude.com/docs/en/about-claude/models/overview

Table 2. Summary of evaluation formats. The original multiple-choice evaluation is used for open-weight models with log-likelihood access, while closed API models are evaluated using generation-based variants.

Category	Tasks	Models	Main Metrics
Multiple-choice	Ko-ARC, Ko-WinoGrande, Ko-LAMBADA	Open only	`acc` / `acc_norm` / `acc_npsq`
Generation	Ko-GSM8K, Ko-IFEval, Ko-EQ-Bench	Open + closed	exact match / task-specific scores
Generative MCQA	Ko-ARC-gen, Ko-WinoGrande-gen, Ko-LAMBADA-gen	Closed only	exact match

(`gpt-5-mini-2025-08-07`) and GPT-5.5 (`gpt-5.5-2026-04-23`) (Singh et al., 2025) through the OpenAI API[5]. Open-weight models are evaluated with the HuggingFace Transformers (Wolf et al., 2020) backend of Language Model Evaluation Harness (`lm-evaluation-harness`) (Biderman et al., 2024). All open-model runs use bfloat16 precision on a single NVIDIA B200 GPU.

We evaluate two groups of open-weight models: Korean-developed LLMs spanning five families (KT Mi:dm 2.0 (KT, 2025), LG EXAONE 4.0 (LG-Research et al., 2025), Kakao Kanana-2 (Kanana, 2025), Naver HyperCLOVAX (Team-HyperCLOVA, 2025), and SKT A.X 4.0 (SKT, 2025)) and widely used multilingual LLMs (Qwen 3.5 (Qwen Team, 2026) and Gemma 4 (Google DeepMind, 2026)). This model set allows direct comparison between Korean-specialized and multilingual model families. The full list of evaluated open-weight models, parameter sizes, and HuggingFace model identifiers is provided in Appendix B.

Evaluation formats. We use the original multiple-choice versions of Ko-ARC, Ko-WinoGrande, and Ko-LAMBADA for open-weight models, since these evaluations require access to token-level log-likelihoods. For closed API models, token-level log-likelihoods are not available. We therefore evaluate closed API models on generative variants of these tasks, where the model is presented with explicitly labeled answer options and instructed to output a single label. Ko-ARC uses A-D labels, while Ko-WinoGrande and Ko-LAMBADA use A-B labels. The final answer is extracted from the generated response with a regular expression over the option labels.

Prompt formatting. For open-weight models, we use plain prompting for all checkpoints and additionally evaluate chat-template prompting for instruction-tuned or chat-oriented checkpoints with a defined chat template. Closed API models are evaluated only with chat-style prompting, since their official APIs do not provide an equivalent plain prompting interface.

[5] `https://developers.openai.com/api/docs`

Generation settings. All generation-based evaluations use greedy decoding. This includes Ko-GSM8K, Ko-IFEval, Ko-EQ-Bench, and the generative multiple-choice variants used for closed API models. The model always selects the highest-probability next token rather than sampling from the output distribution. We disable sampling with `do_sample=False` and set the temperature to 0.0. Each evaluation is run once.

Metrics. Table 2 summarizes the task formats and metrics. For multiple-choice tasks, we report accuracy under raw log-likelihood (`acc`), byte-length normalized log-likelihood (`acc_norm`), and NPSQ-based accuracy (`acc_npsq`). NPSQ (*Normalized Probability Shift by the Question*) compares the model's preference among answer choices conditioned on the question against its preference computed without the question, and counts an example as correct only when the question itself shifts the model toward the correct choice. This separates question-driven evidence from superficial choice preference that arises from lexical, length, or surface-form biases over the answer choices alone (Cho et al., 2026).

For generation tasks, we report task-specific automatic metrics, such as answer exact match for Ko-GSM8K, strict and loose instruction-following accuracy for Ko-IFEval, and the custom Ko-EQ-Bench score. The full evaluation protocol, including stop sequences, prompting templates, answer-extraction rules, and metric definitions, is provided in Appendix C.

4.2. Results

We summarize the main evaluation findings in this section. Complete per-model results for all evaluation settings are reported in Appendix E.

Overall performance. Table 3 reports the best-performing open-weight model for each task under the primary metric used for that task. For MCQA tasks, we use `acc_npsq` as the primary metric, as it accounts for answer-choice priors and better isolates question-driven evidence. For generation tasks, we treat the following as the primary metrics: strict exact match for Ko-GSM8K, prompt-level strict accuracy for Ko-IFEval, and EQ-Bench score

Table 3. Best-performing open-weight model for each task. The metric column indicates the score used to identify the best-performing model.

Task	Primary metric	Prompt format	Best-performing open-weight model	Size	Score
Ko-WinoGrande	acc_npsq	plain	Gemma-4-31B	31B	71.98
		chat	Gemma-4-31B-it	31B	61.17
Ko-LAMBADA	acc_npsq	plain	Kanana-2-30b-a3b-mid-2601	30B	97.87
		chat	EXAONE-4.0-32B	32B	83.68
Ko-ARC-Easy	acc_npsq	plain	Gemma-4-31B	31B	88.13
		chat	Gemma-4-31B-it	31B	80.51
Ko-ARC-Challenge	acc_npsq	plain	Gemma-4-31B	31B	70.61
		chat	Gemma-4-31B-it	31B	65.12
Ko-GSM8K	strict-match	plain	Kanana-2-30b-a3b-instruct-2601	30B	83.02
		chat	Gemma-4-31B-it	31B	82.71
Ko-IFEval	prompt_level_strict_acc	plain	A.X 4.0 Light	7B	60.29
		chat	Gemma-4-31B-it	31B	85.02
Ko-EQ-Bench	eqbench	plain	Qwen3.5-35B-A3B	35B	48.31
		chat	Gemma-4-31B-it	31B	54.98

for Ko-EQ-Bench, following the evaluation conventions of the corresponding source benchmarks where applicable. The leading model differs across tasks and across prompt formatting. In the chat-template setting, Gemma-4-31B-it is the strongest model on most tasks, leading all tasks except Ko-LAMBADA. In the plain prompting setting, however, the best-performing models are more varied: Gemma-4-31B leads most MCQA tasks, Kanana-2-30B variants lead Ko-LAMBADA and Ko-GSM8K, Qwen3.5-35B-A3B leads Ko-EQ-Bench, and the 7B A.X 4.0 Light obtains the best Ko-IFEval score. These results indicate that prompt format affects not only absolute scores but also which model appears strongest for a given task.

Table 4. Mean multiple-choice accuracy of open-weight models under three scoring rules: raw log-likelihood (`acc`), byte-length normalized log-likelihood (`acc_norm`), and NPSQ-based accuracy (`acc_npsq`).

Task	Prompt format	acc	acc_norm	acc_npsq
Ko-WinoGrande	plain	61.69	61.06	59.33
	chat	58.08	57.64	55.73
Ko-LAMBADA	plain	92.49	92.13	87.39
	chat	81.22	80.06	69.20
Ko-ARC-Easy	plain	67.81	69.04	71.21
	chat	64.33	62.04	64.15
Ko-ARC-Challenge	plain	45.63	51.57	52.77
	chat	44.53	50.03	49.52

Scoring differences in MCQA tasks. Table 4 shows that the effect of the scoring rule is task-dependent. NPSQ-based accuracy is lower than raw accuracy on Ko-LAMBADA, especially under chat-template prompting, where the mean score drops from 81.22 to 69.20. Ko-WinoGrande shows a smaller decrease under NPSQ. In constrast, both Ko-ARC subsets often obtain higher scores under `acc_npsq` than under raw accuracy, with the larger difference on Ko-ARC-Challenge. Byte-length normalization has its largest effect on Ko-ARC-Challenge. These results suggest that answer-choice priors affect Ko-LAMBADA most strongly, while the impact of scoring rules varies with the structure of each multiple-choice task.

Effects of prompt formatting. Prompt format has a large effect on open-weight models, but the direction of the effect differs by model family (Appendix E reports the complete results, including plain prompting averages over the chat-template subset). For several chat models, applying the chat template substantially improves generation performance over plain prompting. For example, Gemma 4 and EXAONE-4.0 model families show large gains under chat prompting across Ko-GSM8K, Ko-IFEval, and Ko-EQ-Bench. These cases highlight that evaluations without the intended chat template can substantially underestimate the capabilities of instruction-tuned models.

The reverse pattern appears in the Qwen3.5 series. Models larger than 2B perform well on Ko-GSM8K under plain prompting but drop to almost zero under chat prompting, and flexible answer extraction does not recover the scores. Output inspection suggests that the chat template leads these larger models to produce long English reasoning traces, often without reaching a final Korean answer. In contrast, smaller Qwen3.5 models more often produce short Korean reasoning traces and retain some performance. This pattern suggests that chat-oriented instruction tuning may further reinforce the tendency observed in prior work that multilingual LLMs often rely on English as an internal reasoning language even for non-English tasks (Schut et al., 2025; Zhang et al., 2025), particularly in larger models where such behavior may degrade performance on Korean tasks. More details are provided in Appendix D.

A different pattern appears for some Korean-developed models. Mi:dm 2.0 Base and Kanana-2-30b-a3b-instruct-2601 obtain much higher flexible extract than strict match scores on Ko-GSM8K under chat prompting, indicating that many outputs contain the correct numerical answer but do not follow the expected final answer format. In these cases, the low strict score reflects answer format mismatch rather than complete reasoning failure. Taken together, these results show that prompt format is not a minor implementation detail. It interacts with the model's post-training and can change whether the model follows the intended task, switches to

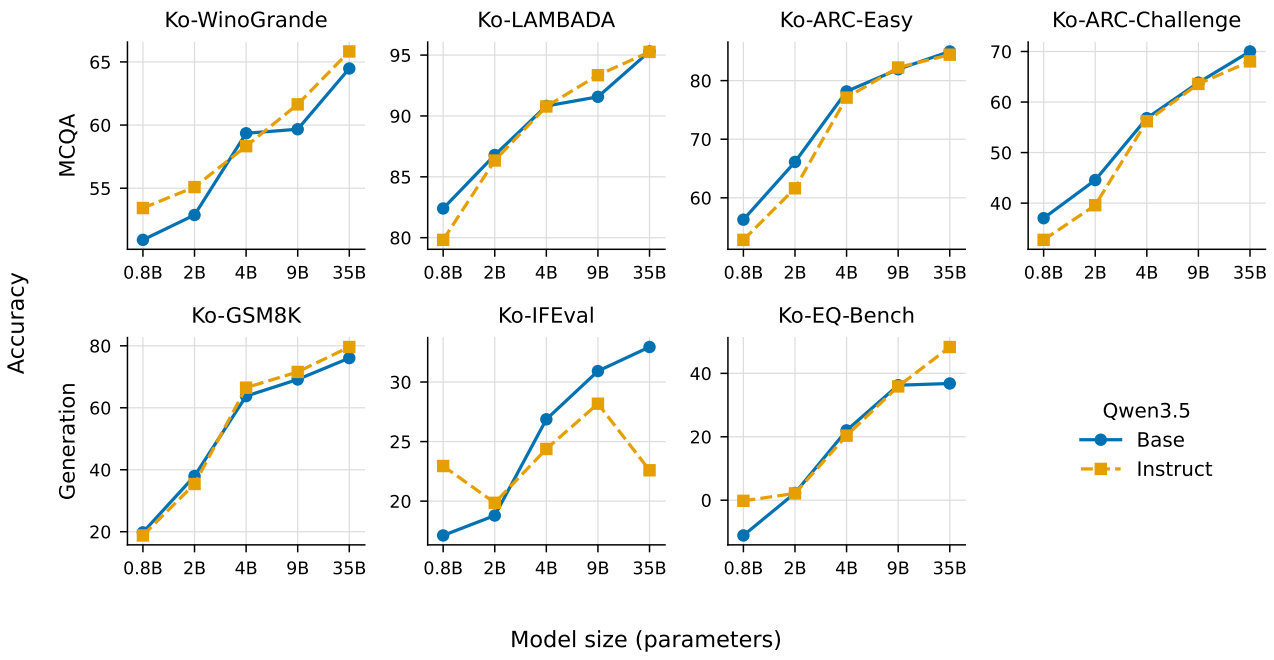

Figure 3. Performance of Qwen3.5 Base (pretrained) and instruction-tuned variants across model sizes under plain prompting. Each panel reports the primary metric used for the corresponding task in Table 3. Larger models generally perform better, while the effect of instruction tuning differs across tasks.

English reasoning on Korean inputs, or fails the required answer format.

Effects of instruction tuning and model size. Figure 3 compares Qwen3.5 Base and instruction variants across model sizes under plain prompting. Larger models generally perform better, especially on Ko-ARC and Ko-GSM8K. However, instruction tuning is not a uniform gain in multilingual model families. In Qwen3.5, the instruction variant improves some scores, such as Ko-GSM8K at larger sizes, but does not consistently outperform the Base model across MCQA tasks or Ko-IFEval. A similar pattern appears in Gemma under plain prompting, where instruction-tuned models often underperform their Base counterparts. On the other hand, the Kanana series shows that later post-training stages improve Ko-GSM8K and lead to stronger overall generation performance. These results suggest that instruction tuning effects depend on what the post-training process is optimized for, including the target language, task format, and prompting style.

Closed API models. Closed API models are reported separately for MCQA because they are evaluated with label-generation variants rather than log-likelihood-based scoring. Within the closed API group, Claude Opus 4.7 and GPT-5.5 are strongest overall: Opus leads most label-generation MCQA variants and Ko-EQ-Bench, while GPT-5.5 leads Ko-GSM8K and Ko-IFEval prompt-level strict accuracy. GPT-5-mini performs well on Ko-GSM8K and Ko-ARC, but is weaker on Ko-LAMBADA and Ko-IFEval. On the shared generation tasks, Claude Opus 4.7 and GPT-5.5 score much higher than the open-weight models on Ko-GSM8K. On Ko-IFEval, however, the best open-weight model (Gemma-4-31B-it) obtains higher scores, and on Ko-EQ-Bench, the comparison depends on the metric. In short, closed API models are strong, but they do not uniformly dominate open-weight models across tasks and metrics. Detailed results are provided in Table 13 in Appendix E.

5. Conclusion

We present a Thunder Korean Benchmark Suite for evaluating foundation models across six capabilities. The suite is constructed through task-level routes that combine expert-reviewed translation and localization, direct Korean construction, and Korean-specific redesign. To our knowledge, few Korean evaluation suites jointly address broad capability coverage, task-specific benchmark construction, differences in model access, and prompt-formatting conditions. For multiple-choice tasks, we use log-likelihood-based scoring and NPSQ-based accuracy for open-weight models, and evaluate closed API models via label generation. For generation tasks, both open-weight and closed-API models are evaluated based on model-generated outputs. We further

compare plain prompting and chat-template prompting for open-weight models, where applicable, while closed-API models are evaluated under chat-style prompting. The results show that Korean benchmark scores are shaped by task construction, scoring rules, prompt format, and model post-training choices. These factors can change both the measured score and the interpretation of model behavior. We therefore suggest that Korean evaluations report not only final scores but also how each task is adapted, how each score is computed, and under which prompting and model-access settings the score is obtained.

Acknowlegments

This work was partially supported by the National Research Foundation of Korea (NRF) under Grant No. RS-2023-00222663 (Center for Optimizing Hyperscale AI Models and Platforms) and under Grant No. A400-20260031, and by the Institute for Information and Communications Technology Promotion (IITP) under Grant No. 2018-0-00581 (CUDA Programming Environment for FPGA Clusters) and No. RS-2025-02304554 (Efficient and Scalable Framework for AI Heterogeneous Cluster Systems), all funded by the Ministry of Science and ICT (MSIT) of Korea. It was also partially supported by the Korea Health Industry Development Institute (KHIDI) under Grant No. RS-2025-25454559 (Frailty Risk Assessment and Intervention Leveraging Multimodal Intelligence for Networked Deployment in Community Care), funded by the Ministry of Health and Welfare (MOHW) of Korea. Additional support was provided by the BK21 Plus Program for Innovative Data Science Talent Education (Department of Data Science, Seoul National University, No. 5199990914569) and the BK21 FOUR Program for Intelligent Computing (Department of Computer Science and Engineering, Seoul National University, No. 4199990214639), both funded by the Ministry of Education (MOE) of Korea. This work was also partially supported by the Advanced GPU Utilization Support Program, funded by the Ministry of Science and ICT (MSIT) of Korea and operated by the National IT Industry Promotion Agency (NIPA). Research facilities were provided by the Institute of Computer Technology (ICT) at Seoul National University.

Impact Statement

This work aims to improve the reliability and reproducibility of Korean foundation model evaluation by providing a benchmark suite with documented construction and evaluation protocols. We expect the suite to support more transparent comparison of Korean and multilingual models, and to make evaluation resources for Korean more accessible to the research community.

We will release the evaluation code under the MIT license. Dataset subsets adapted from existing English benchmarks will be distributed under the licenses of their respective source datasets, including CC-BY for WinoGrande, CC-BY-SA for ARC, MIT for GSM8K and EQ-Bench, and Apache 2.0 for IFEval. Ko-LAMBADA, which is newly constructed from copyright-cleared Korean literary texts, will be released under CC-BY-NC-SA 4.0.

References

ANTHROPIC. System card: Claude haiku 4.5, 2025. URL https://www-cdn.anthropic.com/7aad69bf12627d42234e01ee7c36305dc2f6a970.pdf.

Balepur, N., Ravichander, A., and Rudinger, R. Artifacts or abduction: How do LLMs answer multiple-choice questions without the question? In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10308–10330, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.555. URL https://aclanthology.org/2024.acl-long.555/.

Balepur, N., Rudinger, R., and Boyd-Graber, J. L. Which of these best describes multiple choice evaluation with llms? a) forced b) flawed c) fixable d) all of the above. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3394–3418, 2025.

Bean, A. M., Kearns, R. O., Romanou, A., Hafner, F. S., Mayne, H., Batzner, J., Foroutan, N., Schmitz, C., Korgul, K., Batra, H., Deb, O., Beharry, E., Emde, C., Foster, T., Gausen, A., Grandury, M., Han, S., Hofmann, V., Ibrahim, L., Kim, H., Kirk, H. R., Lin, F., Liu, G. K.-M., Luettgau, L., Magomere, J., Rystrøm, J., Sotnikova, A., Yang, Y., Zhao, Y., Bibi, A., Bosselut, A., Clark, R., Cohan, A., Foerster, J. N., Gal, Y., Hale, S. A., Raji, I. D., Summerfield, C., Torr, P., Ududec, C., Rocher, L., and Mahdi, A. Measuring what matters: Construct validity in large language model benchmarks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026. URL https://openreview.net/forum?id=mdA5lVvNcU.

Biderman, S., Schoelkopf, H., Sutawika, L., Gao, L., Tow, J., Abbasi, B., Aji, A. F., Ammanamanchi, P. S., Black, S., Clive, J., et al. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*, 2024.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G.,

Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., et al. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45, 2024.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. D. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Cho, G., So, Y., and Lee, J. Choices speak louder than questions. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=LzpzC4gd4G.

Chua, L., Ghazi, B., Huang, Y., Kamath, P., Kumar, R., Manurangsi, P., Sinha, A., Xie, C., and Zhang, C. Crosslingual capabilities and knowledge barriers in multilingual large language models. In *Second Conference on Language Modeling*, 2025.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Desai, A. P., Prajapati, R., Ravi, T., Luqman, M., and Yadav, P. Emerging trends in llm benchmarking. In *2024 IEEE International Conference on Big Data (BigData)*, pp. 8805–8807. IEEE, 2024.

Doddapaneni, S., Khan, M. S. U. R., Venkatesh, D., Dabre, R., Kunchukuttan, A., and Khapra, M. M. Cross-lingual auto evaluation for assessing multilingual llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29297–29329, 2025.

Durmus, E., Ladhak, F., and Hashimoto, T. Spurious correlations in reference-free evaluation of text generation. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1443–1454, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.102. URL https://aclanthology.org/2022.acl-long.102/.

Google DeepMind. Gemma 4 model card, 2026. URL https://ai.google.dev/gemma/docs/core/model_card_4.

Guha, N., Nyarko, J., Ho, D., Ré, C., Chilton, A., Chohlas-Wood, A., Peters, A., Waldon, B., Rockmore, D., Zambrano, D., et al. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in neural information processing systems*, 36:44123–44279, 2023.

Ham, J., Choe, Y. J., Park, K., Choi, I., and Soh, H. Kornli and korsts: New benchmark datasets for korean natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 422–430, 2020.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

Hong, S., Kim, S., Son, G., Kim, S., Hong, Y., and Lee, J. From KMMLU-redux to pro: A professional Korean benchmark suite for LLM evaluation. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 19067–19096, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1038. URL https://aclanthology.org/2025.findings-emnlp.1038/.

Jang, M., Kim, D., Kwon, D. S., and Davis, E. Kobest: Korean balanced evaluation of significant tasks. In *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 3697–3708, 2022.

Jin, D., Pan, E., Oufattole, N., Weng, W.-H., Fang, H., and Szolovits, P. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.

Kanana. Release of the smarter and more efficient kanana-2 open-source models. Kakao Tech Blog, December 2025. URL https://tech.kakao.com/posts/804. Original Korean title: "더 똑똑하고 효율적인 Kanana-2 오픈소스 공개". Accessed: 2026-05-02.

Kim, E., Suk, J., Oh, P., Yoo, H., Thorne, J., and Oh, A. Click: A benchmark dataset of cultural and linguistic intelligence in korean. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 3335–3346, 2024.

Kim, H., Kim, D., Kim, J., Lee, S., Kim, Y., and Park, C. Open ko-llm leaderboard2: Bridging foundational and practical evaluation for korean llms. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pp. 266–273, 2025.

Ko, H., Son, G., and Choi, D. Understand, solve and translate: Bridging the multilingual mathematical reasoning gap. In *Proceedings of the 5th Workshop on Multilingual Representation Learning (MRL 2025)*, pp. 78–95, 2025.

KT, T. I. G. Mi:dm 2.0: Korea-centric bilingual language models, 2025. URL https://github.com/K-intelligence-Midm/Midm-2.0?tab=readme-ov-file.

Lee, J., Kim, M., Kim, S., Kim, J., Won, S., Lee, H., and Choi, E. Kornat: Llm alignment benchmark for korean social values and common knowledge. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11177–11213, 2024.

LG-Research, Bae, K., Choi, E., Choi, K., Choi, S. J., Choi, Y., Han, K., Hong, S., Hwang, J., Hwang, T., et al. Exaone 4.0: Unified large language models integrating non-reasoning and reasoning modes. *arXiv preprint arXiv:2507.11407*, 2025.

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

Molfese, F. M., Moroni, L., Gioffré, L., Scirè, A., Conia, S., and Navigli, R. Right answer, wrong score: Uncovering the inconsistencies of llm evaluation in multiple-choice question answering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 18477–18494, 2025.

Paech, S. J. Eq-bench: An emotional intelligence benchmark for large language models, 2023.

Paperno, D., Kruszewski, G., Lazaridou, A., Pham, N.-Q., Bernardi, R., Pezzelle, S., Baroni, M., Boleda, G., and Fernández, R. The lambada dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 1525–1534, 2016.

Park, C., Kim, H., Kim, D., Cho, S., Kim, S., Lee, S., Kim, Y., and Lee, H. Open ko-llm leaderboard: Evaluating large language models in korean with ko-h5 benchmark. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3220–3234, 2024.

Park, S., Moon, J., Kim, S., Cho, W. I., Han, J. Y., Park, J., Song, C., Kim, J., Song, Y., Oh, T., et al. Klue: Korean language understanding evaluation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Pezeshkpour, P. and Hruschka, E. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 2006–2017, 2024.

Phan, L., Gatti, A., Han, Z., Li, N., Hu, J., Zhang, H., Zhang, C. B. C., Shaaban, M., Ling, J., Shi, S., et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

Polo, F. M., Xu, R., Weber, L., Silva, M., Bhardwaj, O., Choshen, L., de Oliveira, A. F., Sun, Y., and Yurochkin, M. Efficient multi-prompt evaluation of llms. *Advances in Neural Information Processing Systems*, 37:22483–22512, 2024.

Qwen Team. Qwen3.5: Towards native multimodal agents, 2026. URL https://qwen.ai/blog?id=qwen3.5.

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Schut, L., Gal, Y., and Farquhar, S. Do multilingual llms think in english? *arXiv preprint arXiv:2502.15603*, 2025.

Sclar, M., Choi, Y., Tsvetkov, Y., and Suhr, A. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *arXiv preprint arXiv:2310.11324*, 2023.

Seo, J., Lee, J., Park, C., Hong, S., Lee, S., and Lim, H.-S. Kocommongen v2: A benchmark for navigating korean commonsense reasoning challenges in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2390–2415, 2024.

Shen, Y., Song, K., Tan, X., Zhang, W., Ren, K., Yuan, S., Lu, W., Li, D., and Zhuang, Y. Taskbench: Benchmarking large language models for task automation. *Advances in*

Neural Information Processing Systems, 37:4540–4574, 2024.

Sindhu, B., Prathamesh, R., Sameera, M., and KumaraSwamy, S. The evolution of large language model: Models, applications and challenges. In *2024 international conference on current trends in advanced computing (ICCTAC)*, pp. 1–8. IEEE, 2024.

Singh, A., Fry, A., Perelman, A., Tart, A., Ganesh, A., El-Kishky, A., McLaughlin, A., Low, A., Ostrow, A., Ananthram, A., et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

SKT, T. A.X 4.0: Foundation model specialized in korean, optimized for enterprise applications, 2025. URL https://github.com/SKT-AI/A.X-4.0.

Son, G., Lee, H., Kim, S., Kim, H., cheol Lee, J., Yeom, J. W., Jung, J., woo Kim, J., and Kim, S. Hae-rae bench: Evaluation of korean knowledge in language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 7993–8007, 2024.

Son, G., Ko, H., and Choi, D. Multi-step reasoning in korean and the emergent mirage. In *Proceedings of the 3rd Workshop on Cross-Cultural Considerations in NLP (C3NLP 2025)*, pp. 10–21, 2025a.

Son, G., Lee, H., Kim, S., Kim, S., Muennighoff, N., Choi, T., Park, C., Yoo, K. M., and Biderman, S. Kmmlu: Measuring massive multitask language understanding in korean. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4076–4104, 2025b.

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on machine learning research*, 2023.

Team-HyperCLOVA. HyperCLOVA X THINK Technical Report, 2025. URL https://arxiv.org/abs/2506.22403.

Wang, H., Zhao, S., Qiang, Z., Xi, N., Qin, B., and Liu, T. Llms may perform mcqa by selecting the least incorrect option. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 5852–5862, 2025.

Wang, X., Hu, C., Ma, B., Rottger, P., and Plank, B. Look at the text: Instruction-tuned language models are more robust multiple choice selectors than you think. In *First Conference on Language Modeling*, 2024a. URL https://openreview.net/forum?id=qHdSA85GyZ.

Wang, Y., Ma, X., Zhang, G., Ni, Y., Chandra, A., Guo, S., Ren, W., Arulraj, A., He, X., Jiang, Z., et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024b.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Wu, Y., Gardner, M., Stenetorp, P., and Dasigi, P. Generating data to mitigate spurious correlations in natural language inference datasets. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2660–2676, 2022.

Xu, Z., Wang, Y., Huang, Y., Chen, X., Zhao, J., Jiang, M., and Zhang, X. Cross-lingual pitfalls: Automatic probing cross-lingual weakness of multilingual large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8254–8284, 2025.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

Zhang, X., Liang, Y., Meng, F., Zhang, S., Huang, K., Chen, Y., Xu, J., and Zhou, J. Think natively: Unlocking multilingual reasoning with consistency-enhanced reinforcement learning. *arXiv preprint arXiv:2510.07300*, 2025.

Zheng, C., Zhou, H., Meng, F., Zhou, J., and Huang, M. Large language models are not robust multiple choice selectors. *arXiv preprint arXiv:2309.03882*, 2023.

Zhou, C., Li, Q., Li, C., Yu, J., Liu, Y., Wang, G., Zhang, K., Ji, C., Yan, Q., He, L., et al. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *International Journal of Machine Learning and Cybernetics*, 16(12):9851–9915, 2025.

Zhou, J., Lu, T., Mishra, S., Brahma, S., Basu, S., Luan, Y., Zhou, D., and Hou, L. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

A. Detailed Correction and Localization Rules

This section describes the dataset-specific rules used during the expert correction and localization stage introduced in Section 3.2. Throughout these stages, we follow two principles: the correct answer should remain unchanged, and the localized item should preserve the difficulty of the original item.

A.1. Source-Side Corrections in English Benchmarks

Some inconsistencies are present in the English source benchmarks rather than being introduced by translation. We correct these cases during the expert correction and localization stage so that the Ko-X dataset reflects the intended task design of the corresponding X dataset rather than source-side artifacts. Representative cases include:

- **ARC-Challenge near-duplicates.** The items `LEAP_8_10365` and `LEAP_2000_8_2` differ only in the surface form "one centimeter" vs. "1 centimeter". We retain one item and remove the other. Other near-duplicate pairs are handled using the same criterion.
- **ARC-Easy unintended numeral.** The question stem "Which of the 27 following is the best use of a robot?" in `MCAS_2005_8_27` contains an unintended "27", which we remove.
- **GSM8K calculator annotations.** GSM8K marks calculator-evaluable expressions with `<<...>>`. Redundant annotations such as `<<8=8>>` are removed, and annotations with incorrect arithmetic are corrected. The original annotation convention is otherwise preserved for compatibility with existing GSM8K evaluation code.

A.2. Ko-GSM8K: Names, Units, and Currency

Names. English given names are replaced with Korean given names sampled from the statistics of high-frequency birth names recorded between 2008 and 2024 in the Korean Supreme Court e-family Register[6]. Name substitutions are kept consistent within each problem so that the same person is referred to by the same Korean name throughout the problem.

Units. Imperial units are converted to metric units. When a numeric value is part of the arithmetic structure of the problem, we use an arithmetically simple metric counterpart rather than an exact conversion (e.g., 10 feet $\rightarrow$ 3 meters rather than 3.048 meters). After each conversion, we recompute the answer to verify that the localized item preserves the intended difficulty.

Currency. U.S. dollars are converted to Korean won using the fixed rate 1 USD = 1,000 KRW. This rate is chosen to preserve the arithmetic structure of the original problems while keeping the quantities natural in Korean. For example, if a problem asks for the cost of three items priced at $15 each, the original calculation is simply $15 \times 3 = 45$. Using a realistic exchange rate such as 1 USD = 1,512 KRW would turn this into $15 \times 1512 \times 3$, adding unintended arithmetic complexity. With the fixed rate, the problem becomes $15000 \times 3 = 45000$, preserving the intended multiplication difficulty. Although this abstracts away from exact exchange-rate variation, the rounded conversion improves arithmetic clarity.

Korean counting expressions. Korean counting expressions are selected according to the noun being counted. For example, "5 pencils" is translated as "연필 5자루", where "자루" is the counting word used with pencil-like objects. We apply the same principle to other noun classes, including people, books, vehicles, and animals, etc.

Everyday items, brands, and places. Everyday items, foods, brands, and place names that are less natural in Korean contexts are replaced with familiar Korean counterparts, provided that the substitution does not change the mathematical structure of the original problem.

A.3. Ko-ARC: Cultural and Geographic Localization

ARC items often contain U.S.-specific contextual details, such as states, national parks, coastlines, animal species, and industrial settings. When these details are not essential to the scientific concept being tested, we replace them with Korean or East Asian counterparts that are more natural in Korean educational contexts. Each substitution is checked against relevant Korean educational and public reference materials. Table 5 summarizes representative substitution rules.

[6] https://stfamily.scourt.go.kr/st/StFrrStatcsView.do?pgmId=090000000025

Table 5. Representative localization rules for Ko-ARC. Each substitution is checked to preserve both the underlying scientific mechanism and the correct answer.

Category	Original context in English ARC	Localized context in Ko-ARC
Locations	U.S. states, national parks, coastlines, and rivers	Korean or East Asian locations with analogous phenomena (e.g., U.S. Gulf coast → West Sea / Yellow Sea; Maine solar-noon example → Baengnyeongdo Island). Globally familiar references, such as the Himalayas, are retained when they are already standard in Korean science education.
Climate and seasonality	Hurricanes; jet-stream examples set in the continental U.S.	Comparable weather and seasonal phenomena in the Northwest Pacific (e.g., *hurricanes* → *typhoons*; jet-stream examples rewritten using Korean winter-weather patterns; daylight-duration comparisons are rewritten using Korean cities with appropriate latitude differences).
Species and organisms	Black bear; warbler migration; Gulf sturgeon	Ecologically comparable species that are familiar in Korean educational contexts (e.g., Asiatic black bear in Jirisan Mountain; cranes migrating between Siberia or Manchuria and the Korean peninsula; salmon as a river-sea migratory species).
Industry and energy	Kentucky coal-transport context; U.S. 2003 electricity generation mix	Korean settings with comparable industrial or energy-related roles (e.g., Jecheon-si cement freight corridor; Korean 2022 electricity generation mix).
Units and measurements	Imperial units; city-by-city daylight comparisons	Metric units; numerical values adjusted when needed to preserve the scientific relation; city pairs reselected within Korea.
Everyday objects and cultural examples	U.S.-specific household or cultural items used to frame the question	Korean counterparts that make the question natural in Korean while preserving the scientific framing and the correct answer.

A.4. Ko-EQ-Bench: Emotion Labels and Honorific Usage

Emotion-label mapping. Some English emotion labels encode distinctions that do not map cleanly onto a single Korean label. To ensure consistent annotation across items, we use a documented mapping table for emotion labels. When a direct label mapping does not preserve the intended meaning in context, we apply an item-specific revision. For example, ”Victimized” is mapped to ”피해의식”, which captures a perceived sense of ”being wronged” rather than ”the literal state of having been harmed”. Idiomatic expressions are also translated by preserving their contextual meaning rather than their literal form. For example, ”haunted” is translated as ”계속 떠오른다” when it refers to ”a thought or memory that repeatedly comes to mind”, rather than as the literal ghost-related meaning ”귀신이 나오는”.

Honorific usage. Direct translation can produce repeated address forms in which a Korean name is followed by the polite suffix ”씨” throughout a dialogue. Since such repetition can sound unnatural or overly explicit in Korean, we reduce it where appropriate, while preserving the relationship between speakers and the emotional tone of the original item.

A.5. Ko-WinoGrande: Names and Everyday Objects

English personal names are replaced with Korean given names (e.g., Jessica → 지희). The substitutions are kept consistent within each item and checked to preserve the original coreference structure. Everyday objects that are uncommon or unnatural in Korean contexts are replaced with familiar Korean counterparts when the replacement does not change the commonsense relation required by the item (e.g., toaster oven → 전자레인지(microwave) as Koreans typically don’t use toaster ovens). Sentence structure is also adjusted to avoid excessive repetition of names, which is more marked in Korean than in English.

A.6. Ko-IFEval: Hybrid Adaptation and Korean-Specific Verifiers

Construction subsets. Ko-IFEval is constructed through a hybrid strategy. Items with indices 1,000-3,757 form the translated subset, where original IFEval prompts are adapted through the four-stage pipeline described in Section 3.2. For these items, we translate and localize the prompt while preserving the original instruction-following constraint whenever it

Table 6. Instruction-type inventory in the original IFEval and Ko-IFEval. The "Changes" column summarizes Ko-IFEval-specific additions and adaptations.

Instruction group	IFEval	Ko-IFEval	Changes
Keywords	4	6	Alphabet-based letter constraints are separated from Korean choseong(초성; leading consonant)-based constraints; a keyword-substitution check is added for Korean prompts.
Length Constraints	4	6	Character-count checks are extended to distinguish whitespace-inclusive and whitespace-exclusive counting, which is necessary for Korean spacing conventions.
Detectable Format	6	7	A multiple-choice formatting check is added.
Language	1	1	No structural change; the required response language can be Korean or English.
Detectable Content	2	2	No structural change.
Combination	2	2	No structural change.
Change Cases	3	3	No structural change.
Start with / End with	2	2	No structural change.
Punctuation	1	1	No structural change.
Total	**25**	**30**	+5 new or adapted instruction types.

can be evaluated reliably in Korean. Localization includes replacing culturally unfamiliar genres with Korean counterparts. For example, an instruction asking for an English "limerick" is rewritten as a request for "5행시", a Korean five-line poem written from a given word or phrase. We also convert verifier arguments, such as comparison relations and target strings, so that the automatic checker can evaluate Korean outputs directly.

Items with indices 10,001-10,300 form the Korean-specific subset. For these items, authors write Korean prompts from scratch using public-domain Korean source material, including presidential speeches. We define additional instruction types that reflect Korean writing conventions and implement verifier code for checking these constraints automatically.

Instruction-type expansion. Table 6 summarizes how the instruction-type inventory is expanded from 25 types in the original IFEval to 30 types in Ko-IFEval. The expansion is concentrated in instruction groups where direct transfer from English is insufficient, especially keyword constraints, length constraints, and detectable formatting. Other instruction groups are retained without structural change.

Verifier arguments. For Korean-specific instruction types, each item includes Korean-aware arguments in the `kwargs` field. For example, items with the `detectable_format:number_highlighted_sections` instruction specify `num_highlights` together with a Korean comparison relation such as "이상(greater than or equal to)" or "미만(less)". Length-constraint items also specify whether `num_letters` should be counted with or without whitespace. The verifier code reads these fields directly, allowing the original IFEval scoring framework to be extended to Ko-IFEval with a small compatibility layer.

B. Open-Weight Models Evaluated

Table 7 lists all open-weight models included in our evaluation, grouped by provider. We report the model size and the exact Hugging Face checkpoint used for evaluation. The Korean-developed group spans five model families released by Korean companies, while the multilingual group includes two widely used model families with multilingual capabilities.

C. Detailed Evaluation Protocol

Common settings. All evaluations are conducted with `lm-evaluation-harness`. We use the test split for every task and run each evaluation once. For generation-based tasks, we use greedy decoding with `do_sample=False` and `temperature=0.0`. Because sampling is disabled, sampling-related parameters such as `top_p` and random seeds are not used.

Models. Closed API models are evaluated through their official APIs. We use Claude Haiku 4.5 (`claude-haiku-4-5-20251001`) and Claude Opus 4.7 (`claude-opus-4-7`) through the Anthropic API,

Table 7. Open-weight models evaluated. Sizes are reported in billions of parameters. For mixture-of-experts models, we also include the active-parameter shorthand reported by the developer, such as `30B-A3B`. All checkpoints are accessed through the Hugging Face Hub.

Provider	Model	Size	Hugging Face checkpoint
Korean-developed LLMs			
KT	Mi:dm 2.0 Mini	2.3B	`K-intelligence/Midm-2.0-Mini-Instruct`
	Mi:dm 2.0 Base	11.5B	`K-intelligence/Midm-2.0-Base-Instruct`
LG	EXAONE-4.0-1.2B	1.2B	`LGAI-EXAONE/EXAONE-4.0-1.2B`
	EXAONE-4.0-32B	32B	`LGAI-EXAONE/EXAONE-4.0-32B`
Kakao	Kanana-2-30b-a3b-base-2601	30B-A3B	`kakaocorp/kanana-2-30b-a3b-base-2601`
	Kanana-2-30b-a3b-mid-2601	30B-A3B	`kakaocorp/kanana-2-30b-a3b-mid-2601`
	Kanana-2-30b-a3b-instruct-2601	30B-A3B	`kakaocorp/kanana-2-30b-a3b-instruct-2601`
Naver	HyperCLOVAX-SEED-Text-Instruct-0.5B	0.5B	`HyperCLOVAX-SEED-Text-Instruct-0.5B`
	HyperCLOVAX-SEED-Text-Instruct-1.5B	1.5B	`HyperCLOVAX-SEED-Text-Instruct-1.5B`
SKT	A.X 4.0 Light	7B	`skt/A.X-4.0-Light`
Multilingual LLMs			
Qwen	Qwen3.5-0.8B-Base	0.8B	`Qwen/Qwen3.5-0.8B-Base`
	Qwen3.5-0.8B	0.8B	`Qwen/Qwen3.5-0.8B`
	Qwen3.5-2B-Base	2B	`Qwen/Qwen3.5-2B-Base`
	Qwen3.5-2B	2B	`Qwen/Qwen3.5-2B`
	Qwen3.5-4B-Base	4B	`Qwen/Qwen3.5-4B-Base`
	Qwen3.5-4B	4B	`Qwen/Qwen3.5-4B`
	Qwen3.5-9B-Base	9B	`Qwen/Qwen3.5-9B-Base`
	Qwen3.5-9B	9B	`Qwen/Qwen3.5-9B`
	Qwen3.5-27B	27B	`Qwen/Qwen3.5-27B`
	Qwen3.5-35B-A3B-Base	35B-A3B	`Qwen/Qwen3.5-35B-A3B-Base`
	Qwen3.5-35B-A3B	35B-A3B	`Qwen/Qwen3.5-35B-A3B`
Google	Gemma-4-E2B	5B	`google/gemma-4-E2B`
	Gemma-4-E2B-it	5B	`google/gemma-4-E2B-it`
	Gemma-4-E4B	8B	`google/gemma-4-E4B`
	Gemma-4-E4B-it	8B	`google/gemma-4-E4B-it`
	Gemma-4-26B-A4B	26B-A4B	`google/gemma-4-26B-A4B`
	Gemma-4-26B-A4B-it	26B-A4B	`google/gemma-4-26B-A4B-it`
	Gemma-4-31B	31B	`google/gemma-4-31B`
	Gemma-4-31B-it	31B	`google/gemma-4-31B-it`

and GPT-5-mini (`gpt-5-mini-2025-08-07`) and GPT-5.5 (`gpt-5.5-2026-04-23`) through the OpenAI API. Open-weight models are evaluated using the Hugging Face Transformers backend of `lm-evaluation-harness`; each model is identified by its exact Hugging Face checkpoint. Open-weight model inference uses bfloat16 precision on a single NVIDIA B200 GPU with 180 GB memory.

Stop sequences. For the generative variants of Ko-ARC, Ko-WinoGrande, and Ko-LAMBADA, generation stops at one of `\n`, `</s>`, or `<|im_end|>`. For Ko-GSM8K, generation stops at 문제:, `</s>`, or `<|im_end|>`. Ko-IFEval uses no explicit stop sequence. Ko-EQ-Bench uses no explicit stop sequence. For Ko-EQ-Bench, generation is bounded only by the maximum generation length. We use 80 tokens for open-weight models, but increase the limit to 1024 tokens for closed API models because their responses may include additional hidden or explicit reasoning tokens before the final emotion-score lines.

Prompt formatting conditions. For open-weight models, we distinguish between plain prompting and chat-template prompting. Plain prompting is used for all open-weight checkpoints. For instruction-tuned or chat-oriented checkpoints with a defined chat template, we additionally evaluate chat-template prompting using the `--apply_chat_template` option in `lm-evaluation-harness`. This option formats each benchmark prompt according to the checkpoint-specific chat template by adding model-specific role markers, special tokens, or generation prefixes. When few-shot examples are used, the demonstrations and the test instance are represented as alternating user and model turns.

For example, for a Gemma instruction-tuned checkpoint, a task prompt containing Korean benchmark-specific fields such as 질문: and 답변: is wrapped as:

```
<bos><|turn>user
```

Table 8. Detailed task configuration. Original multiple-choice tasks are evaluated only for open-weight models with log-likelihood access. Generative multiple-choice variants are used for closed API models. For Ko-EQ-Bench, the maximum generation length is 80 tokens for open-weight models and 1024 tokens for closed API models.

Task	Few-shot	Output type	Max gen. toks	Metric
Ko-ARC-Easy	5	multiple choice	–	`acc` / `acc_norm` / `acc_npsq`
Ko-ARC-Challenge	5	multiple choice	–	`acc` / `acc_norm` / `acc_npsq`
Ko-WinoGrande	0	multiple choice	–	`acc` / `acc_norm` / `acc_npsq`
Ko-LAMBADA	0	multiple choice	–	`acc` / `acc_norm` / `acc_npsq`
Ko-GSM8K	5	generate-until	2048	`strict-match` / `flexible-extract`
Ko-IFEval	0	generate-until	1280	prompt-level and instruction-level strict/loose accuracy
Ko-EQ-Bench	0	generate-until	80 / 1024	`eqbench` / `percent_parseable`
Ko-ARC-gen	5	generate-until	512	`exact_match`
Ko-WinoGrande-gen	0	generate-until	512	`exact_match`
Ko-LAMBADA-gen	0	generate-until	512	`exact_match`

```
[benchmark prompt]
<turn|>
<|turn>model
```

Here, the Korean field labels are part of the benchmark prompt, whereas the turn markers and special tokens are added by the chat template.

Closed API models are evaluated only through their official chat-style APIs. Since these APIs do not expose tokenizer-level chat-template control or a comparable plain prompting interface, we cannot evaluate closed models under the plain prompting condition. Closed-model results should therefore be interpreted as chat-style prompting results, and compared most directly with the open-weight chat-template condition rather than the open-weight plain condition.

Prompting and answer extraction. For original multiple-choice tasks, open-weight models are evaluated using log-likelihood-based scoring. For Ko-ARC, the model receives the question prompt, and each answer choice is scored as a candidate continuation. For Ko-WinoGrande and Ko-LAMBADA, each candidate is inserted into the blank and scored by log-likelihood comparison.

For closed API models, we use generative multiple-choice variants because token-level log-likelihoods are not available. Ko-ARC-Easy-gen and Ko-ARC-Challenge-gen (generation versions) present four labeled options and require the model to output a single label:

```
질문: {question}
A. {choices.text[0]}
B. {choices.text[1]}
C. {choices.text[2]}
D. {choices.text[3]}
반드시 A, B, C, D 중 하나의 문자로만 답하세요.
정답:
```

The final answer is extracted with the following regular expression:

```
(?i)(?:정답\s*[:：]?\s*)?([A-D])(?:\s*(?:번|입니다|이에요|가)|[\.)\]:：,\s]|$)
```

Ko-WinoGrande-gen presents two labeled options and requires the model to output either A or B:

```
문장의 빈칸에 들어갈 말로 더 자연스러운 선택지를 고르세요.
문장: {sentence}
A. {option1}
B. {option2}
반드시 A 또는 B 중 하나의 문자로만 답하세요.
정답:
```

Ko-LAMBADA-gen uses the same two-choice format, with the masked passage provided as the sentence field:

```
문장의 빈칸에 들어갈 말로 더 자연스러운 선택지를 고르세요.
문장: {text}
A. {option_A}
B. {option_B}
반드시 A 또는 B 중 하나의 문자로만 답하세요.
정답:
```

For Ko-WinoGrande-gen and Ko-LAMBADA-gen, the final answer is extracted with the same regular expression, except that the valid label range is A-B:

```
(?i)(?:정답\s*[:：]?\s*)?([A-B])(?:\s*(?:번|입니다|이에요|가)|[\.)\]:：,\s]|$)
```

If no valid label is matched, the response is counted as incorrect.

For Ko-GSM8K, the prompt format is `문제: {question}\n답:`. We use two answer extractors: a strict extractor that looks for an explicit final-answer marker and a flexible extractor that falls back to the final numeric expression in the response. During answer normalization, commas, currency markers such as `$` and "원", text before `####`, and a trailing period are ignored.

Ko-IFEval uses the `prompt` field directly and is evaluated with deterministic instruction-following verifiers. Ko-EQ-Bench also uses the `prompt` field directly; the model is expected to output four lines of the form `emotion: score`.

Ko-EQ-Bench also uses the`prompt` field directly. The model is expected to output four emotion-score pairs, one per line, in the form `emotion: score`. The Ko-EQ-Bench scorer extracts lines matching `([\w가-힣]+):\s*(\d+)`, and counts the response as parseable only when exactly four emotion-score pairs are found and all four predicted emotion labels match the reference emotion labels. If these conditions are not satisfied, the example receives `eqbench`=0 and `percent_parseable`=0.

Metric definitions. For multiple-choice tasks, `acc` denotes the fraction of examples for which the correct option has the highest log-likelihood. `acc_norm` applies length normalization to the per-choice log-likelihoods before comparison. `acc_npsq` uses Normalized Probability Shift by the Question (NPSQ; Cho et al., 2026). For a question-related input Q, choice-related input C, and answer candidate x, NPSQ is defined as

$$\mathrm{NPSQ}(Q, C, x) = \frac{\log P(x \mid Q, C) - \log P(x \mid C)}{-\log P(x \mid C)}.$$

The prediction is the option with the highest NPSQ score. This scoring rule measures the relative increase in the model's preference for each choice after the question is included, compared with a question-free baseline that scores the choices without the question. It therefore separates question-driven evidence from choice-only preferences such as choice-length bias or surface-form familiarity.

For Ko-GSM8K, we report two exact-match variants: `strict-match` and `flexible-extract`. The `strict-match` metric counts an example as correct only when the response contains an explicit final-answer marker of the form `#### number` and the extracted number matches the gold answer after normalization. The `flexible-extract` metric uses the same matching rule but extracts the final numeric expression in the response when the explicit final-answer marker is absent.

For Ko-IFEval, we report four instruction-following accuracies. Prompt-level strict accuracy counts a prompt as correct only when all verifiable instructions in that prompt are satisfied under the strict verifier. Instruction-level strict accuracy is the fraction of individual verifiable instructions that are satisfied. Prompt-level loose accuracy and instruction-level loose accuracy use the same two aggregation levels, but apply the loose verifier, which allows common surface variations in the response before checking whether each instruction is followed.

For Ko-EQ-Bench, we report two metrics: `eqbench` and `percent_parseable`. The model is instructed to output four emotion-score pairs in the form `emotion: score`, with integer scores on a 0-10 scale. `percent_parseable` is the percentage of examples for which the model output can be parsed into exactly four emotion-score pairs and all four

predicted emotion labels match the reference labels. For parseable outputs, `eqbench` measures how closely the predicted emotion intensity scores match the reference scores. For each matched emotion, we compute the absolute difference between the predicted and reference scores and apply the same scaled penalty function used by the EQ-Bench scorer. The final example-level score is converted to a 0-100 scale, where higher values indicate closer agreement with the reference emotion intensities. Unparseable outputs receive `eqbench`=0 and `percent_parseable`=0 before aggregation, so the reported `eqbench` score reflects both emotion-score accuracy and output format compliance.

D. Qwen3.5 Behavior under Chat Prompting

We inspect representative Ko-GSM8K outputs from Qwen3.5-9B and Qwen3.5-0.8B under chat prompting. The example uses a simple arithmetic problem about selling duck eggs. Both models have enough information to solve the problem, and the correct answer is 18,000. However, their generation behavior differs. Qwen3.5-0.8B directly solves the problem in Korean and reaches the required final answer format. In contrast, Qwen3.5-9B begins with a long English reasoning trace and does not produce the final answer in the recorded output. This example illustrates that the Qwen3.5 failure under chat prompting is not simply a lack of mathematical ability but is related to the language and length of the generated reasoning trace.

Qwen3.5-0.8B. Qwen3.5-0.8B produces a short Korean solution and reaches the final answer.

> 보리의 하루는 16개의 알을 낳습니다. 아침 식사로 3개, 머핀으로 4개, 나머지는 직거래 장터에서 16 - 3 - 4 = 9개의 알을 판매합니다.
>
> 매일 직거래 장터에서 벌이는 돈은 9개 x 2000원 = ¡¡9*2000=18000¿¿18,000원입니다.
>
> #### 18000

Qwen3.5-9B. Qwen3.5-9B begins solving the same problem in English and produces a long reasoning trace. In the recorded output, the model does not reach the final answer.

> Thinking Process:
>
> 1. Analyze the Request: The user is presenting a math word problem in Korean and expects a solution in a specific format ...
>
> 2. Analyze the Problem:
> * Duck lays 16 eggs per day.
> * Duck eats 3 eggs for breakfast.
> * Duck makes muffins with 4 eggs for friends.
> * Remaining eggs are sold at the direct market.
> * Selling price: 2000 won per egg.
>
> 3. Step-by-Step Calculation:
> * Total eggs laid: 16
> * Eggs eaten: 3
> * Eggs for muffins: 4
> * Eggs remaining for sale: $16 - 3 - 4$
> * Calculation: $16 - 3 = 13$, $13 - 4 =$

E. Full Experimental Results

This section reports the complete per-model results for all evaluation settings. Tables 9 and 10 report open-model MCQA results under plain prompting and chat template prompting, respectively, with `acc`, `acc_norm`, and `acc_npsq`. Tables 11 and 12 report open-model generation results under the two prompt formats. Table 13 reports the closed API model results, including the label generation variants of MCQA tasks and the generation tasks.

Table 9. Open-weight multiple-choice results under plain prompting. The chat-template subset average uses the same model subset as the chat-template prompting results in Table 10. The highest score in each metric column is highlighted in red.

Models	Parameter size	ko-winogrande			ko-lambada			ko-arc-easy			ko-arc-challenge		
		acc	acc _norm	acc _npsq	acc	acc _norm	acc _npsq	acc	acc _norm	acc _npsq	acc	acc _norm	acc _npsq
HyperCLOVAX-SEED-Text-Instruct-0.5B	0.5B	55.72	55.88	54.62	93.30	92.95	87.72	58.71	58.67	61.66	32.39	38.73	38.05
HyperCLOVAX-SEED-Text-Instruct-1.5B	1.5B	56.83	57.06	54.70	90.02	89.22	84.17	60.35	61.45	59.51	34.53	40.62	38.13
A.X 4.0 Light	7B	64.96	63.38	62.75	96.59	96.32	93.13	78.41	78.87	79.46	55.87	59.73	61.27
Mi:dm 2.0 Mini	2.3B	61.25	60.62	58.80	96.76	96.54	93.48	75.55	76.60	78.11	48.33	54.84	58.70
Mi:dm 2.0 Base	11.5B	66.22	65.04	63.69	98.32	98.18	96.72	81.57	82.66	85.23	58.78	64.18	65.55
EXAONE-4.0-1.2B	1.2B	51.93	51.78	51.86	74.68	74.01	67.58	36.79	36.79	40.07	22.62	28.02	25.79
EXAONE-4.0-32B	32B	60.06	59.83	58.64	94.41	93.93	90.33	78.33	80.01	81.10	55.10	59.55	62.90
Kanana-2-30b-a3b-base-2601	30B	67.88	67.48	66.77	98.76	98.67	97.78	78.37	79.00	81.90	54.33	60.93	63.15
Kanana-2-30b-a3b-mid-2601	30B	69.06	67.88	65.67	98.67	98.58	97.87	79.42	78.75	82.11	54.41	60.50	63.50
Kanana-2-30b-a3b-instruct-2601	30B	71.27	70.40	68.90	98.45	98.36	97.47	80.05	80.30	81.36	58.61	63.24	63.41
Qwen3.5-0.8B-Base	0.8B	53.35	53.99	50.91	89.36	88.38	82.40	51.14	51.98	56.27	32.99	36.68	37.02
Qwen3.5-0.8B	0.8B	55.01	55.17	53.43	87.94	86.79	79.82	47.43	49.41	52.82	29.14	33.16	32.73
Qwen3.5-2B-Base	2B	57.93	57.06	52.88	93.61	93.08	86.79	60.35	62.46	66.12	38.48	44.47	44.56
Qwen3.5-2B	2B	57.77	57.46	55.09	92.68	91.89	86.34	55.98	58.38	61.62	33.16	38.65	39.59
Qwen3.5-4B-Base	4B	62.35	61.01	59.35	96.14	95.70	90.82	73.36	74.54	78.16	49.02	56.30	56.81
Qwen3.5-4B	4B	61.41	61.25	58.33	95.34	94.95	90.78	72.05	73.36	77.10	47.90	55.10	56.21
Qwen3.5-9B-Base	9B	65.04	63.85	59.67	96.45	96.50	91.57	77.74	79.63	81.94	54.84	61.78	63.84
Qwen3.5-9B	9B	64.64	63.85	61.64	96.67	96.36	93.35	77.10	78.75	82.24	53.64	61.10	63.58
Qwen3.5-27B	27B	71.51	70.64	68.59	97.92	97.83	95.88	80.64	83.33	84.89	58.36	64.61	67.27
Qwen3.5-35B-A3B-Base	35B	69.30	67.72	64.48	97.38	97.38	95.30	81.19	83.50	84.98	60.24	66.67	70.01
Qwen3.5-35B-A3B	35B	67.17	66.93	65.83	97.47	97.25	95.26	80.60	82.87	84.43	58.53	66.84	68.04
Gemma-4-E2B	5B	59.91	58.80	58.25	96.05	95.39	91.66	65.78	66.84	69.19	40.96	46.96	48.67
Gemma-4-E2B-it	5B	51.22	50.99	50.67	70.20	70.02	60.98	39.94	40.70	40.91	26.39	30.51	29.22
Gemma-4-E4B	8B	66.30	65.75	63.85	97.65	97.47	94.32	73.95	75.55	78.24	50.21	55.18	59.98
Gemma-4-E4B-it	8B	51.78	51.14	49.65	76.19	75.26	64.26	49.24	49.87	50.51	27.42	34.70	33.42
Gemma-4-26B-A4B	27B	70.56	68.98	67.64	98.36	98.09	96.67	80.93	82.24	86.03	56.98	64.01	67.10
Gemma-4-26B-A4B-it	27B	51.14	51.54	49.49	82.31	82.13	67.36	56.78	57.16	58.29	35.99	40.96	41.56
Gemma-4-31B	31B	74.74	73.17	71.98	98.63	98.63	97.07	83.54	85.94	88.13	61.18	67.78	70.61
Gemma-4-31B-it	31B	52.64	52.09	52.49	81.91	82.00	67.36	51.26	52.53	52.78	32.82	39.67	39.59
average		61.69	61.06	59.33	92.49	92.13	87.39	67.81	69.04	71.21	45.63	51.57	52.77
chat-template subset average		59.59	59.17	57.73	90.06	89.67	84.00	64.49	65.65	67.34	42.75	48.57	49.17

Table 10. Open-weight multiple-choice results under chat-template prompting. The highest score in each metric column is highlighted in red.

Models	Parameter size	ko-winogrande			ko-lambada			ko-arc-easy			ko-arc-challenge		
		acc	acc _norm	acc _npsq	acc	acc _norm	acc _npsq	acc	acc _norm	acc _npsq	acc	acc _norm	acc _npsq
HyperCLOVAX-SEED-Text -Instruct-0.5B	0.5B	55.49	55.09	54.85	83.99	83.19	74.99	55.56	55.60	57.74	31.19	36.59	35.48
HyperCLOVAX-SEED-Text -Instruct-1.5B	1.5B	57.46	56.35	55.09	83.86	83.55	78.76	57.91	59.72	58.88	33.25	40.19	37.45
A.X 4.0 Light	7B	58.72	59.12	57.77	86.56	85.81	78.49	75.46	77.15	77.36	50.64	56.56	56.47
Mi:dm 2.0 Mini	2.3B	59.67	58.49	54.78	83.55	81.82	66.83	72.39	73.70	75.25	45.59	52.19	51.76
Mi:dm 2.0 Base	11.5B	64.33	63.06	60.62	86.43	85.06	68.65	77.53	74.45	79.59	54.67	58.61	61.70
EXAONE-4.0-1.2B	1.2B	50.28	50.20	51.22	76.28	73.84	62.48	58.12	49.41	53.54	34.19	41.30	38.73
EXAONE-4.0-32B	32B	56.12	57.70	57.14	88.60	88.12	83.68	75.04	74.54	77.10	51.76	56.04	57.41
Kanana-2-30b-a3b-instruct-2601	30B	61.09	61.25	60.46	89.00	88.60	82.79	76.39	74.03	75.25	54.67	58.61	58.10
Qwen3.5-0.8B	0.8B	51.78	51.70	50.83	73.66	71.80	61.20	45.88	47.18	48.49	29.05	34.96	33.59
Qwen3.5-2B	2B	55.41	55.01	52.80	76.63	74.63	63.77	54.04	54.25	57.03	33.16	40.10	38.48
Qwen3.5-4B	4B	56.83	55.41	53.04	76.67	75.26	61.73	47.98	41.16	44.15	36.50	43.10	43.70
Qwen3.5-9B	9B	57.30	56.04	55.96	77.56	76.23	61.38	51.14	43.27	46.30	38.73	46.02	44.73
Qwen3.5-27B	27B	59.35	58.33	57.77	82.97	81.29	70.51	55.77	47.90	54.55	47.13	51.59	52.70
Qwen3.5-35B-A3B	35B	60.22	59.91	56.83	79.51	77.78	66.12	53.87	44.74	45.67	43.79	46.70	48.50
Gemma-4-E2B-it	5B	54.14	53.59	51.70	74.37	73.57	62.44	67.30	65.78	67.26	44.90	49.96	47.73
Gemma-4-E4B-it	8B	59.12	58.96	58.09	80.04	79.38	68.60	75.63	75.00	76.98	52.79	59.13	56.90
Gemma-4-26B-A4B-it	27B	61.88	62.19	52.96	79.25	79.16	61.55	77.61	78.96	79.13	57.58	63.58	62.81
Gemma-4-31B-it	31B	66.22	65.19	61.17	83.10	82.00	71.62	80.35	79.92	80.51	62.04	65.30	65.12
average		58.08	57.64	55.73	81.22	80.06	69.20	64.33	62.04	64.15	44.53	50.03	49.52

Table 11. Open-weight generation results under plain prompting. The chat-template subset average uses the same model subset as the chat-template prompting results in Table 12. The highest score in each metric column is highlighted in red.

Models	Parameter size	ko-gsm8k		ko-ifeval				ko-eq-bench	
		strict -match	flexible -extract	prompt _level _strict _acc	inst _level _strict _acc	prompt _level _loose _acc	inst _level _loose _acc	eqbench	percent _parseable
HyperCLOVAX-SEED-Text-Instruct-0.5B	0.5B	4.09	3.79	12.49	25.99	14.63	29.06	5.01	70.18
HyperCLOVAX-SEED-Text-Instruct-1.5B	1.5B	9.40	7.89	21.64	36.73	23.54	39.44	31.56	71.93
A.X 4.0 Light	7B	78.85	78.92	60.29	74.36	65.16	77.56	44.06	76.61
Mi:dm 2.0 Mini	2.3B	57.85	61.71	19.62	29.13	24.26	32.96	9.39	30.41
Mi:dm 2.0 Base	11.5B	81.73	81.80	28.30	40.70	31.87	43.83	39.52	70.76
EXAONE-4.0-1.2B	1.2B	4.02	7.13	2.26	5.16	2.38	5.23	-0.11	4.68
EXAONE-4.0-32B	32B	78.24	79.38	37.22	52.13	40.55	54.77	45.24	73.10
Kanana-2-30b-a3b-base-2601	30B	69.67	69.83	13.44	22.44	15.22	24.53	28.65	76.02
Kanana-2-30b-a3b-mid-2601	30B	76.73	76.65	17.60	33.31	19.74	35.82	24.86	51.46
Kanana-2-30b-a3b-instruct-2601	30B	83.02	83.17	30.44	49.48	36.15	54.08	42.03	74.85
Qwen3.5-0.8B-Base	0.8B	19.79	20.32	17.12	30.24	17.96	31.43	-11.12	64.91
Qwen3.5-0.8B	0.8B	18.80	19.11	22.95	39.93	26.99	43.90	-0.20	14.04
Qwen3.5-2B-Base	2B	37.98	38.59	18.79	35.12	19.74	36.38	2.26	64.33
Qwen3.5-2B	2B	35.41	35.86	19.86	36.79	20.57	37.56	2.14	22.22
Qwen3.5-4B-Base	4B	63.76	64.75	26.87	43.07	30.08	45.51	22.01	65.50
Qwen3.5-4B	4B	66.49	67.25	24.38	41.53	25.57	42.79	20.36	54.39
Qwen3.5-9B-Base	9B	69.14	69.83	30.92	48.43	35.79	52.06	36.26	66.08
Qwen3.5-9B	9B	71.57	72.56	28.18	46.20	29.73	47.46	35.88	64.91
Qwen3.5-27B	27B	30.33	31.92	21.76	39.23	22.24	39.65	29.15	42.11
Qwen3.5-35B-A3B-Base	35B	76.04	77.71	32.94	50.38	38.41	54.15	36.80	64.33
Qwen3.5-35B-A3B	35B	79.61	80.67	22.59	40.21	23.07	40.63	48.31	69.59
Gemma-4-E2B	5B	17.97	18.12	15.34	29.97	15.93	30.73	-12.18	76.02
Gemma-4-E2B-it	5B	5.91	10.01	4.76	10.94	4.88	11.01	10.50	64.91
Gemma-4-E4B	8B	47.38	48.07	18.43	32.40	19.38	33.03	19.17	64.91
Gemma-4-E4B-it	8B	58.61	54.21	12.01	21.05	12.13	21.19	7.73	31.58
Gemma-4-26B-A4B	27B	58.53	59.36	17.12	31.43	18.43	33.31	14.38	69.59
Gemma-4-26B-A4B-it	27B	29.95	31.08	15.58	29.13	16.65	29.97	16.47	33.92
Gemma-4-31B	31B	74.30	75.36	19.86	35.12	21.17	36.86	33.82	67.25
Gemma-4-31B-it	31B	42.31	44.28	13.79	26.41	13.91	26.48	7.76	23.39
average		49.91	50.67	21.60	35.76	23.66	37.63	20.33	56.00
chat-template subset average		46.46	47.26	22.12	35.84	24.13	37.64	21.93	49.64

Table 12. Open-weight generation results under chat-template prompting. The highest score in each metric column is highlighted in red.

Models	Parameter size	ko-gsm8k		ko-ifeval				ko-eq-bench	
		strict -match	flexible -extract	prompt _level _strict _acc	inst _level _strict _acc	prompt _level _loose _acc	inst _level _loose _acc	eqbench	percent _parseable
HyperCLOVAX-SEED-Text-Instruct-0.5B	0.5B	3.56	6.07	17.48	32.33	18.79	34.50	18.56	74.27
HyperCLOVAX-SEED-Text-Instruct-1.5B	1.5B	12.51	11.90	22.24	36.31	24.38	39.23	34.12	72.52
A.X 4.0 Light	7B	82.18	82.34	73.72	83.07	76.10	85.09	44.11	76.61
Mi:dm 2.0 Mini	2.3B	66.04	67.32	53.98	65.78	57.67	68.43	13.70	36.26
Mi:dm 2.0 Base	11.5B	32.60	84.08	70.16	80.63	74.79	83.76	37.20	62.57
EXAONE-4.0-1.2B	1.2B	46.10	47.92	50.30	61.53	53.27	63.76	16.50	59.06
EXAONE-4.0-32B	32B	70.51	76.27	74.79	83.69	80.62	87.60	46.72	76.61
Kanana-2-30b-a3b-instruct-2601	30B	44.88	86.35	76.58	84.95	81.45	88.15	44.11	74.27
Qwen3.5-0.8B	0.8B	23.05	24.41	32.46	48.43	34.13	50.38	13.97	76.02
Qwen3.5-2B	2B	43.06	44.35	44.59	61.19	48.75	64.46	24.79	76.61
Qwen3.5-4B	4B	0.15	2.12	20.93	38.33	21.52	38.82	0.00	0.00
Qwen3.5-9B	9B	0.00	2.73	20.57	38.12	20.81	38.54	0.00	0.00
Qwen3.5-27B	27B	0.00	2.73	21.05	38.68	21.40	38.96	0.00	0.00
Qwen3.5-35B-A3B	35B	0.08	3.26	21.17	38.82	21.52	39.16	0.00	0.00
Gemma-4-E2B-it	5B	33.59	54.13	66.83	78.54	72.06	82.44	40.25	76.02
Gemma-4-E4B-it	8B	63.76	67.55	74.79	84.18	78.72	86.97	53.17	76.02
Gemma-4-26B-A4B-it	27B	77.86	80.67	80.50	87.74	83.35	89.90	52.88	76.61
Gemma-4-31B-it	31B	82.71	84.38	85.02	90.24	87.75	92.27	54.98	76.02
average		37.92	46.03	50.40	62.92	53.17	65.13	27.50	54.97

Table 13. Closed API model results across all benchmark tasks under chat-style prompting.

model	ko -winogrande -gen	ko -lambada -gen	ko-arc -easy -gen	ko-arc -challenge -gen	ko-gsm8k		ko-ifeval				ko-eqbench	
	exact _match	exact _match	exact _match	exact _match	strict -match	flexible -extract	prompt _level _strict _acc	inst _level _strict _acc	prompt _level _loose _acc	inst _level _loose _acc	eqbench	percent _parseable
Claude Haiku 4.5	78.45	91.93	97.77	94.17	77.18	81.05	72.18	82.65	81.93	88.99	50.51	74.85
Claude Opus 4.7	93.21	98.67	99.03	97.26	96.29	96.29	80.38	87.46	84.3	90.03	60.52	76.02
GPT-5-mini	76.64	65.41	97.26	95.29	81.50	93.78	38.76	35.96	39.00	36.17	47.13	76.02
GPT-5.5	91.08	96.90	98.91	97.43	96.36	96.44	81.57	84.53	82.64	85.23	60.20	76.02