BenchMAX: A Comprehensive Multilingual Evaluation Suite for Large Language Models

Anonymous ACL submission

Abstract

Existing multilingual benchmarks focus primarily on language understanding tasks. There is a lack of benchmarks to measure comprehensive critical capabilities of large language models (LLMs) across diverse languages, including instruction following, reasoning, code generation, and long context understanding. To bridge this gap, we develop BenchMAX, a multi-way multilingual benchmark to evaluate LLMs' general abilities across many languages. BenchMAX consists of high-quality data samples annotated by native annotators in 17 languages covering 10 diverse tasks. Extensive experiments on BenchMAX reveal uneven utilization of core capabilities across languages, emphasizing the performance gaps that scaling model size alone does not resolve. BenchMAX serves as a comprehensive multilingual evaluation platform, providing a promising test bed to promote the development of multilingual language models. The dataset and code will be publicly accessible.

1 Introduction

Large Language Models (LLMs; OpenAI et al., 2024; Gemini, 2024; DeepSeek-AI et al., 2024) have displayed remarkable proficiency across a wide range of tasks, mainly because they excel in instruction following, reasoning, long context understanding, code generation, and so on (Ouyang et al., 2022; Cobbe et al., 2021; Su et al., 2024; Roziere et al., 2023; Lu et al., 2024; Sun et al., 2024). Inherently, these capabilities are language-agnostic. The numerical outcome remains consistent regardless of whether one learns the arithmetic expression $1 + 1 = 2$ in English or Chinese. Similarly, when it comes to coding tasks, using English or Chinese instructions does not alter the fundamental logic of the code. However, numerous empirical studies have shown that LLMs' multilingual performance is quite unbalanced when handling the same tasks (Shi et al., 2023; Zhu et al., 2024; Qi et al., 2023) across different languages.

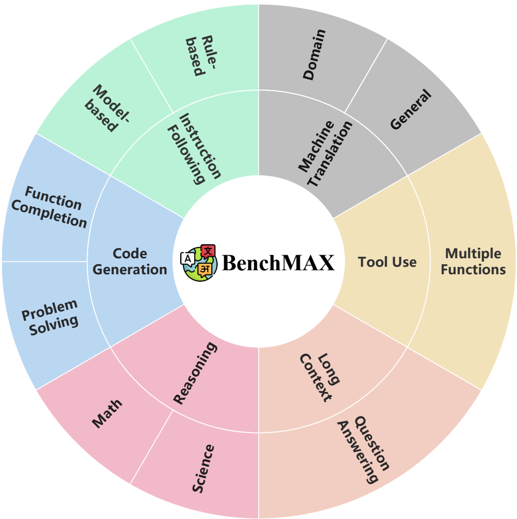


Figure 1: BenchMAX evaluates 6 capabilities of LLMs on 10 diverse tasks across 17 languages.

However, current benchmarks (Hendrycks et al., 2021; Lai et al., 2023; Singh et al., 2024; Wang et al., 2024a) do not support comprehensive testing of the language-agnostic abilities of LLMs, particularly in low-resource languages, for several reasons. Tasks like XWinograd (Muennighoff et al., 2023) and XStoryCloze (Lin et al., 2022), based on multiple-choice formats, do not fully evaluate the generative capacities of LLMs. Additionally, the limited language overlap across existing benchmarks poses challenges in assessing LLM performance in diverse languages. Recently, P-MMEval (Zhang et al., 2024) is proposed as a multilingual multitask benchmark, with the majority of its tasks still following a multiple-choice format. While it includes assessments like MGSM (Shi et al., 2023) and MIFEVAL that cover partial language-agnostic capabilities, this narrow focus still leaves a significant gap between research evaluation and real-world applications.

To tackle this problem, we develop a comprehensive, multi-way, and challenging multilingual evaluation suite, called BenchMAX, to help the community better analyze and improve the language-agnostic capabilities of LLMs. Covering 17 languages[1], BenchMAX not only includes a broader range of language families but also emphasizes the diversity of writing systems across languages. Meanwhile, BenchMAX highlights diverse advanced capabilities including instruction following, code generation, long context understanding, reasoning, tool use, and translation. For evaluating each capability, we include one or two related tasks as shown in Figure 1. Domain translation, a by-product of data construction, poses a new challenge for LLMs by necessitating fine-grained control and domain-specific terminology understanding over the translation process.

To ensure high quality, we design an annotation framework to optimize the dataset quality with human efforts and LLM feedback. The process involves translating data from English to other languages using machine translation systems, post-editing each sample by three native annotators with multiple iterations across most tasks, and picking the final translation version using a strong LLM that involves swapping sample positions for debiasing (Wang et al., 2024b; Li et al., 2024).

Popular multilingual LLMs are evaluated on BenchMAX, revealing that language notably influences language-agnostic capabilities of existing LLMs. Interestingly, simply increasing the parameters can boost average performance on these tasks but does not universally reduce the performance gap across languages. Moreover, compared to general translation, domain translation not only poses new challenges for LLMs but also requires new evaluation metrics. The main contributions can be summarized as follows:

- We develop a comprehensive, multi-way multilingual benchmark across **17** languages for evaluating **6** crucial capabilities on **10** diverse tasks.
- We propose a pipeline for curating high-quality multilingual datasets, involving both human annotation and LLM-as-a-judge.
- We evaluate popular multilingual LLMs on BenchMAX, and the related analyses provide a further understanding of the language-agnostic capabilities.

2 Related Work

Prior to the era of LLMs, most multilingual benchmarks are designed to evaluate discriminative models and take the form of classification tasks, such as XNLI (Conneau et al., 2018), XCOPA (Ponti et al., 2020), XCSQA (Talmor et al., 2019), etc. (Lin et al., 2022; Muennighoff et al., 2023) However, due to their limited complexity and lack of format diversity, these tasks become less practical. Recently, MGSM (Shi et al., 2023) has become the most frequently used dataset in papers and reports from leading LLM teams (Dubey et al., 2024; Gemini, 2024; OpenAI, 2024), which measures the mathematical reasoning capability across 11 languages. Another widely used multilingual benchmark is the translated version of MMLU (Hendrycks et al., 2021; Lai et al., 2023; Singh et al., 2024), which contains knowledge-intensive tasks. However, due to the lack of a unified dataset version, scores are often difficult to compare between studies. Moreover, recent analyses have revealed that MMLU contains numerous ground truth errors (Gema et al., 2024), obscuring the accurate evaluation. More recently, INCLUDE (Romanou et al., 2024) has been proposed to evaluate multilingual regional knowledge, lacking assessment of language-agnostic capabilities. To address these limitations, our work focuses more on language-agnostic capabilities and includes more tasks such as reasoning and code generation. Furthermore, our benchmark incorporates a broader range of tasks to evaluate LLMs multilingual capabilities more comprehensively compared to previous aggregated benchmarks, such as SeaEval (Wang et al., 2024a) and P-MMEval (Zhang et al., 2024). Importantly, all translated samples except the long context data in our benchmark are post-edited by native annotators.

3 Benchmark Construction

We extend the evaluation of the critical capabilities of LLMs into multilingual scenarios. To ensure sufficient linguistic diversity, we select 16 non-English languages (§ 3.1). Meanwhile, 10 diverse tasks evaluating 6 crucial capabilities are chosen to facilitate comprehensive assessment (§ 3.2). Subsequently, we introduce a rigorous pipeline (§ 3.3) that incorporates both human annotators and LLMs to obtain a high-quality benchmark.

[1] The 17 languages include English, Spanish, French, German, Russian, Bengali, Japanese, Thai, Swahili, Chinese, Telugu, Arabic, Korean, Serbian, Czech, Hungarian, and Vietnamese.

Language	ISO	Language Family	Script System	Language	ISO	Language Family	Script System
Hungarian	hu	Uralic	Latin	Serbian	sr	Indo-European	Serbian Cyrillic
Vietnamese	vi	Austroasiatic		Korean	ko	Koreanic	Hangul / Chosŏn'gŭl
Spanish	es	Indo-European		Japanese	ja	Japonic	Mixed scripts of Chinese Characters and Hiragana, Katakana
Czech	cs			Arabic	ar	Afro-Asiatic	Arabic alphabet
French	fr			Thai	th	Kra–Dai	Thai
German	de			Swahili	sw	Niger–Congo	Latin
Russian	ru		Cyrillic	Chinese	zh	Sino-Tibetan	Chinese Characters
Bengali	bn		Bengali–Assamese	Telugu	te	Dravidian	Telugu

Table 1: Besides English, BenchMAX supports 16 non-English languages, covering a wide range of language families and script systems.

Capability	Category	Dataset	# Samples	Metric	Capability	Category	Dataset	# Samples	Metric
Instruction Following	Rule-based	IFEval	429	Accuracy	Code Generation	Function Completion	Humaneval+	164	Pass@1
	Model-based	m-ArenaHard	500	Win Rate		Problem Solving	LiveCodeBench_v4	713	
Reasoning	Math	MGSM	250	Exact Match	Translation	General	Flores+TED+WMT24	[1012, 4049]	spBLEU
	Science	GPQA	448			Domain	Annotated data above	2781	
Tool Use	Multiple Functions	Nexus	318	Accuracy	Long Context Modeling	Question Answering	RULER	800	Exact Match

Table 2: Selection of core capabilities and details of task data. For IFEval, we filter out all language-specific instructions, thus remaining 429 samples. For Nexus, we only adopt the standardized_queries subset which contains 318 samples. For general translation datasets, the number of samples may vary in different translation directions, according to the number of parallel samples in TED and WMT24. The datasets of the model-based instruction following task and math reasoning are expanded from existing multilingual datasets, while others are translated from English datasets.

3.1 Language Selection

BenchMAX supports 17 languages to cover diverse language families and writing systems (Table 1).

3.2 Capabilities Selection

LLMs have demonstrated proficiency in understanding tasks such as text classification and sentiment analysis, but their capabilities transcend understanding. We construct tasks to evaluate following intrinsic capabilities in multilingual settings:

- **Instruction Following:** The capability to follow instructions is evaluated by two distinct tasks with different evaluation paradigms: rule-based and model-based assessment. For the rule-based task, we translate IFEval (Zhou et al., 2023) from English to other languages, while we expand m-ArenaHard (Dang et al., 2024; Li et al., 2024) to languages we select for the model-based task.
- **Reasoning:** The reasoning capability is assessed through intricate scenarios including math and natural science (physics, chemistry, and biology) problems. We expand MGSM (Shi et al., 2023) and GPQA (Rein et al., 2023) to 17 languages for the math reasoning and science reasoning tasks.
- **Code Generation:** We mainly consider Python code generation in two settings, function completion and programming problem solving. We translate Humaneval+ (Liu et al., 2024; Chen et al., 2021) and LiveCodeBench_v4 (Jain et al., 2024) from English to other languages.
- **Long Context Modeling:** We evaluate the ability to extract evidence from lengthy documents through question-answering tasks with long documents (128k tokens). We build this task based on RULER (Hsieh et al., 2024), and translate haystacks, needles, and QA pairs.
- **Tool Use:** We assess the ability to correctly select and invoke a single function from multiple options in response to user queries. We translate the queries in Nexus (Srinivasan et al., 2023) to other languages, but leave the APIs in English.
- **Translation:** Translation converts text between languages while preserving meaning. In addition to standard tasks like Flores, TED, and WMT (Costa-jussà et al., 2022; Cettolo et al., 2012; Kocmi et al., 2024), we introduce the Domain Translation task, a by-product of the BenchMAX construction. It challenges models to translate specialized terminology and determine whether specific segments should be translated.

The information of the curated datasets, sample sizes, and evaluation metrics is provided in Table 2. More details can be found in Appendix A.

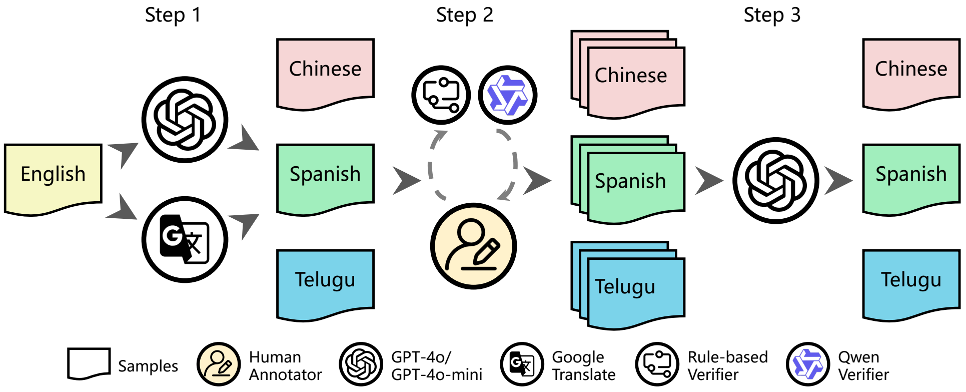


Figure 2: The construction process involves three steps: Step 1) translating data from English to non-English; Step 2) post-editing each sample by three human annotators; Step 3) selecting the final translation version.

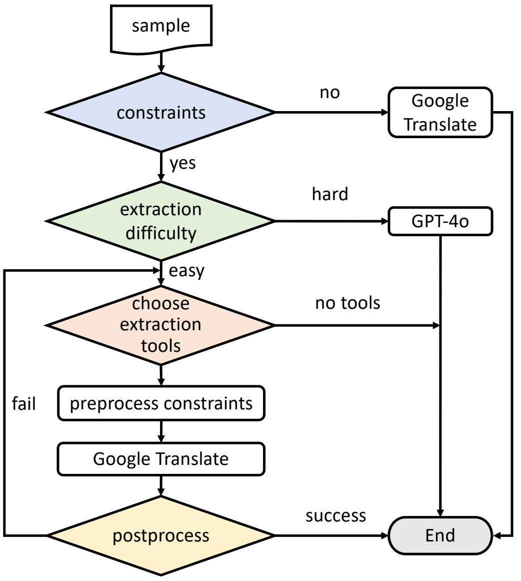


Figure 3: Flow chart illustrating the constraint extraction and machine translation pipeline in the first step of our benchmark construction.

3.3 Construction

The way to obtain BenchMAX consists of three steps, as shown in Figure 2: 1) translate data from English to non-English by machines; 2) post-edit each sample by three native annotators; 3) pick the final translation version by GPT-4o-mini.

Step 1: Translating data from English to selected non-English languages by machine translation systems. We select between specialized translation models such as Google Translate, and LLM-based ones like GPT-4o, based on constraint extractability. As illustrated in Figure 3, if the data contains hard-to-extract constraints, we prompt GPT-4o to translate it and satisfy the constraints. Otherwise, we use Google Translate along with extraction tools. Extraction tools include methods for extracting translated keywords by enclosing source keywords with special symbols, and for preserving source constraints by replacing constraints with placeholders before translation and restoring them afterwards.

Note that in cases where existing multilingual datasets are available, such as MGSM and m-ArenaHard, we extend them to include the supported languages by translating the English data, to minimize additional effort.

Step 2: Post-editing each sample by three distinct native annotators in almost all tasks. To ensure high-quality data, we employ a rigorous multi-round annotation and verification pipeline: 1) Each sample is given to three native annotators who are proficient in English and their native language. Considering the specialized nature of datasets like Science reasoning, annotators are required to hold at least a Bachelor's degree. 2) Two automatic verifiers - rule-based verifiers and model-based verifiers - are used to assess the quality of human annotation. Rule-based verifiers ensure the satisfaction of constraints for certain tasks, such as the rule-based instruction following task. For model-based verifiers, we utilize the GEMBA-SQM prompt and employ Qwen2.5-72B-Instruct, a powerful multilingual model, to estimate the quality of translations. Along with providing an overall score, the model offers detailed explanations of translation errors as feedback to annotators. Samples that do not pass the rule-based verifier or score below a predefined threshold are identified as failed, and refined in subsequent iterations. Each manually annotated dataset undergoes at least three iterations.

Step 3: Selecting the final translation version by LLMs. Initailly, a fourth annotator uninvolved in

Model	Size	Instruction Following		Code Generation		Reasoning		Long Context	Tool Use	Translation			
		Rule-based	Model-based	Func Compl.	Prob. Solving	Math	Science	Question Answering	Multi Func.	General		Domain	
										En-X	X-En	En-X	X-En
InternLM2.5	7B	45.7	1.9	45.4	10.3	37.4	20.6	37.5	53.2	12.7	20.2	34.4	54.0
	20B	51.9	3.3	51.2	14.4	42.9	24.0	-	26.6	14.9	19.7	34.9	53.9
Aya-Expanse	8B	51.2	6.4	33.8	7.8	50.8	26.2	-	41.1	21.5	26.8	45.6	51.6
	32B	61.9	12.4	52.0	15.8	66.7	27.7	-	59.8	25.2	32.8	54.8	62.3
Gemma2	9B	63.0	9.8	53.9	16.6	72.0	23.9	-	61.4	27.2	33.2	57.5	61.9
	27B	62.4	18.0	66.7	24.6	75.3	26.7	-	64.7	30.4	34.5	64.8	66.2
Llama3.1	8B	62.6	4.3	52.9	14.1	63.4	23.8	68.3	45.0	24.6	29.8	53.9	62.9
R1-Distill-Llama3.1	8B	49.7	3.5	62.8	23.8	46.9	28.1	-	37.2	12.2	20.8	13.5	23.1
Llama3.1	70B	76.2	13.2	69.7	29.8	79.7	35.8	57.4	44.3	31.1	**35.1**	64.5	**68.2**
Llama3.3	70B	**85.2**	17.0	74.0	34.7	83.8	42.6	50.4	42.5	31.5	33.6	63.5	65.0
R1-Distill-Llama3.3	70B	78.0	26.6	**84.6**	54.8	82.8	46.1	-	62.1	26.0	33.0	47.6	45.2
Qwen2.5	7B	65.9	8.5	68.2	24.7	63.4	27.6	53.5	48.9	16.6	25.6	46.4	60.0
R1-Distill-Qwen2.5	7B	46.7	3.0	69.3	37.3	56.1	28.4	-	27.7	6.8	16.3	17.0	27.3
Qwen2.5	32B	78.1	17.3	75.8	42.7	77.7	37.7	79.4	66.7	22.7	30.5	54.2	65.4
R1-Distill-Qwen2.5	32B	67.3	19.2	80.6	54.4	77.3	37.0	-	60.4	20.3	28.5	37.1	37.7
Qwen2.5	72B	80.8	36.9	78.6	45.5	77.8	39.4	80.6	61.8	25.8	33.3	60.4	66.9
DeepSeek-V3*	671B	83.9	**59.8**	83.2	**60.4**	**84.2**	**47.4**	**85.2**	69.2	**33.9**	34.5	**70.3**	67.8
GPT-4o-mini	-	79.1	21.9	78.7	37.0	76.9	34.1	82.1	**70.9**	30.3	33.9	67.7	67.6

Table 3: Performance comparison across models on BenchMAX tasks, averaged over 17 languages. Bold numbers indicate the best performance in each column. "Func Compl." refers to Function Completion, "Prob. Solving" to Problem Solving, and "Multi Func." to Multiple Functions scenarios where models must select and call one function from multiple options. Models without results on the long context task do not support 128K context length. * DeepSeek-V3 is a 671B MoE model, with 37B activated for each token.

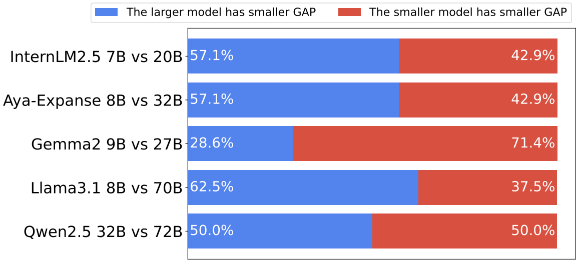


Figure 4: Larger models do not consistently have a smaller GAP. Each row shows proportions of tasks where the larger model achieves a smaller GAP versus where the smaller model performs better.

the previous process selects the final version from the three revised outputs. Interestingly, they show a strong position bias, frequently choosing the first annotation. This likely stems from the consistently high quality of translations, making differences negligible.

Due to the high cost of human debiasing (since three translations cover all six permutations), we use GPT-4o-mini—a strong, multilingual LLM—to select the final translation. In particular, following Li et al. (2024), we adapt the LLM-Judge system instruction (see Appendix G) to suit pairwise translation evaluation. We shuffle the three translations, run two battles, and determine a winner in each through position-swapped judgments. The final winner is chosen by pitting the initial winner against the third translation.

4 Experimental Results

4.1 Evaluation Setup

We mainly focus on post-trained multilingual models and evaluate both open-source and proprietary language models[2], including Llama3.1 (Dubey et al., 2024), Qwen2.5 (Qwen Team, 2024), Gemma2 (Team et al., 2024), InternLM2.5 (Cai et al., 2024), Aya-Expanse (Dang et al., 2024), DeepSeek-R1-Distill-Llama (Guo et al., 2025), DeepSeek-R1-Distill-Qwen, DeepSeek-V3 (DeepSeek-AI et al., 2024), and GPT-4o-mini (OpenAI, 2024). Detailed descriptions of models and the inference configuration can be found in Appendix C & D.

4.2 Multilingual Benchmark Results

Table 3 shows the overall average performance of each model on each multilingual task. More detailed results are in Appendix H.

Model scaling improves overall multilingual performance while language disparities persist. As shown in Table 3, larger models consistently demonstrate enhanced multilingual capabilities across all domains, with few exceptions. However, the performance gap between English and non-English languages does not invariably diminish. We define GAP as the average performance

[2] Unless otherwise specified, all models discussed in this paper are post-trained versions.

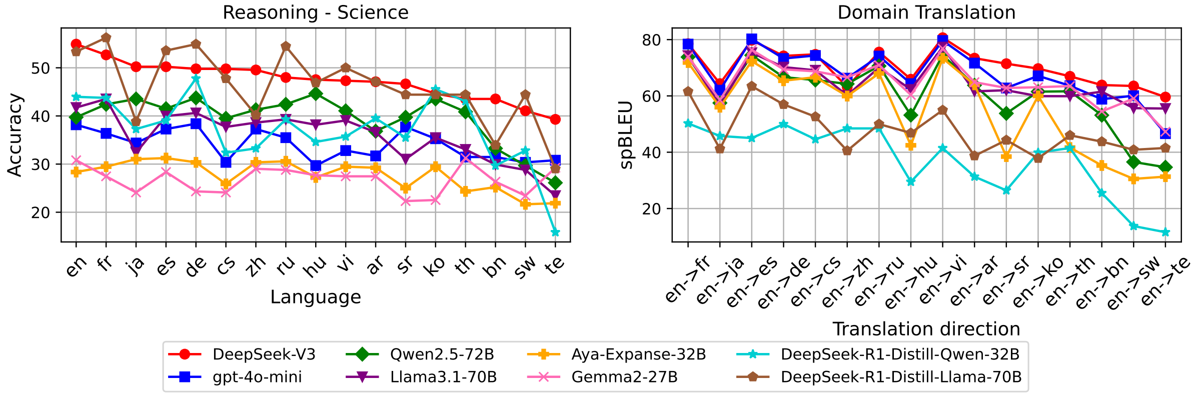


Figure 5: Taking two tasks as examples, models exhibit unbalanced multilingual capabilities. We show performance of several models on the science reasoning task and the domain translation task across different languages.

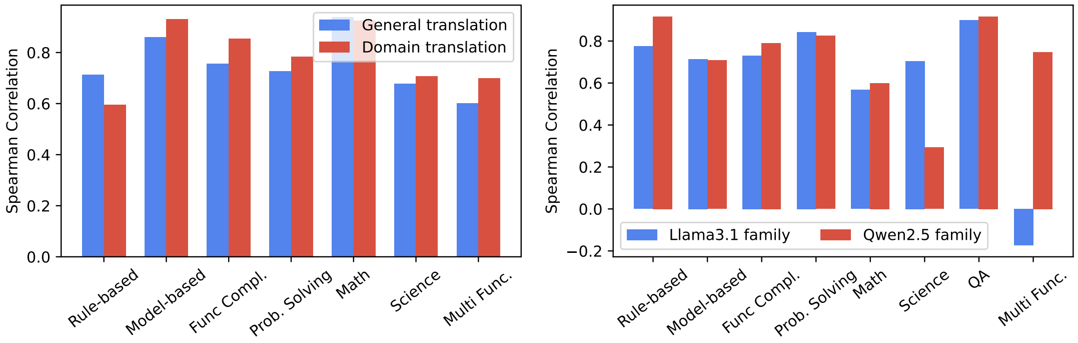


Figure 6: **Left:** The translation performance is positively correlated with other multilingual performance. Spearman Correlations are computed between the performance on general/domain translation and the specific task. **Right:** Models in the same family have similar language performance pattern. We compute the Spearman Correlations between the performance of two models (Llama3.1 8B vs 70B, Qwen2.5 7B vs 72B) across different languages.

gap between English and other languages:

$$GAP = \frac{\sum_{l \neq \text{en}} \max(s(\text{en}) - s(l), 0)}{n-1},$$

where $s(l)$ denotes the score on the task with language l, and n is the number of languages including English. As shown in Figure 4, when comparing models of different sizes, the proportion of larger models achieving smaller GAPs only slightly exceeds 0.5 for most model families. Gemma2-9B achieves smaller GAPs than Gemma2-27B on most tasks. These findings suggest that while scaling model size effectively improves overall multilingual performance, additional strategies may be needed to address the performance disparities across languages.

The effective utilization of language-agnostic capabilities remains challenging in multilingual contexts. The left plot of Figure 5 illustrates that models' reasoning capabilities vary significantly across languages, typically excelling in high-resource languages. This disparity can be attributed to the fact that multilingual task execution depends not only on language-agnostic reasoning but also on language-specific capabilities such as comprehension and generation. Therefore, when operating in weak languages, it becomes difficult for a model to fully leverage its language-agnostic capabilities. Interestingly, we observe an unexpected pattern where certain models excel in specific non-dominant languages compared to English on some tasks. For example, Qwen2.5 demonstrates superior performance in Korean over English on the science reasoning task. We hypothesize that Qwen2.5's training data includes a relatively high proportion of Korean content in scientific or reasoning-related domains. This counter-intuitive phenomenon merits further study.

Model performance exhibits systematic bias towards high-resource languages. As shown in Figure 5, the performance curves of most models exhibit significant fluctuations across languages. High-resource languages such as French and Chinese consistently outperform low-resource lan-

Model	Rule-based	Func Compl.	Understanding
Qwen2.5-7B	No.1	No.1	No.4
Llama3.1-8B	No.3	No.3	No.3
Aya-Expanse-8B	No.4	No.4	No.2
Gemma2-9B	No.2	No.2	No.1

Table 4: Rankings of the models in generation tasks in BenchMAX differ from that in understanding tasks, indicating the importance of both types of tasks. The ranking in understanding tasks is from Dang et al. (2024).

	Llama3.1-70B			Qwen2.5-72B		
Translated by	GT	4o-mini	Ours	GT	4o-mini	Ours
Rule-based	66.9	53.5	**76.2**	71.5	57.2	**80.8**
Func Compl.	47.8	68.2	**69.7**	50.4	75.5	**78.6**
Science	33.7	35.1	**35.8**	36.9	37.8	**39.4**
Multi Func.	23.0	43.7	**44.3**	26.7	61.3	**61.8**

Table 5: Our pipeline provides a more accurate assessment of the multilingual performance, compared to naive translations by Google Translate(GT) and GPT-4o-mini(4o-mini), respectively.

guages like Telugu, Swahili, and Bengali. This pattern can be partially attributed to development strategies - models like Aya-Expanse are not specifically optimized for the full range of languages in our evaluation. Unexpectedly, Gemma2 exhibits relatively balanced performance across most tasks (Figure 7), despite not being explicitly marketed as a multilingual model.

Translation capabilities exhibit a positive correlation with other evaluated capabilities. We analyze the relationship between model's English-to-X translation capability and other capabilities using Spearman correlation coefficients (the left panel of Figure 6). When calculating correlations between domain-specific translation performance and task performance, we exclusively use data from the corresponding domains. The analysis reveals that domain-specific translation performance generally exhibits stronger correlations with task performance compared to general translation capabilities. A notable exception is that in the rule-based instruction-following task, we observe an inverse scaling effect: larger LLMs produce lower-quality translations compared to their smaller counterparts. We find that larger LLMs are more likely to execute instructions rather than strictly perform translation, known as prompt injection.

Models within the same family exhibit consistent performance patterns across languages. We calculate Spearman correlation coefficients to analyze the performance similarity between models of the same family (excluding R1-distilled models) across different languages for each task. As shown in the right panel of Figure 6, models within the same family show strong correlations across various tasks, with most correlation coefficients exceeding 0.7.

R1-distilled models exhibit enhanced multilingual reasoning and code generation capabilities, but some other capabilities, especially translation, are noticeably degraded. As illustrated in Table 3, the performance of R1-Distill-Llama3.3-70B is comparable to DeepSeek-V3 in reasoning and code generation tasks, and is stronger than Llama-3.3-70B-Instruct. However, other capabilities like instruction following of 7B/8B models exhibit degradation to some extent. They tend to generate repeated tokens in the reasoning process when using non-English languages. The translation capabilities of both large and small distilled models decline dramatically. In addition to repeated generation, we also observe a frequent phenomenon of code-switching in translations.

5 Analysis

5.1 Models rank differently in understanding and generation tasks

Prior work like Aya-Expanse (Dang et al., 2024) relies on conventional understanding tasks such as XCOPA and XWinograd for multilingual evaluation. With these metrics, Gemma2-9B achieves the best performance, followed by Aya-Expanse-8B, Llama3.1-8B, and Qwen2.5-7B. However, our evaluation through BenchMAX reveals a different pattern: Qwen2.5-7B demonstrates superior multilingual capabilities on generation tasks, while Aya-Expanse models show notably weaker performance on code generation tasks, as shown in Table 4. This discrepancy highlights the importance of comprehensive evaluation frameworks that incorporate both understanding and generation tasks to accurately assess multilingual capabilities of LLMs.

5.2 Our pipeline provides a more accurate assessment of models' performance

We naively translate a subset of tasks from English to other 16 languages by Google Translate and GPT-4o-mini, and then evaluate two models using this task data. We directly translate sources by Google Translate as it doesn't support constraints, and use appropriate prompts with constraints to request GPT-4o-mini. The results in Table 5 show

Metric	Translation Model	Reasoning - Math				Reasoning - Science				Code generation - Prob. Solving			
		zh	de	sw	te	zh	de	sw	te	zh	de	sw	te
spBLEU	Gemma2-27B	40.0	51.4	38.2	29.2	80.6	84.8	66.2	57.5	85.5	78.5	76.2	52.3
	Llama3.1-70B	35.2	54.4	36.6	35.0	71.8	84.9	64.0	65.3	84.8	78.5	75.5	56.7
	Qwen2.5-72B	37.7	50.1	13.5	12.6	77.0	79.4	41.2	40.9	84.6	73.0	48.8	42.1
TER	Gemma2-27B	36.2	32.1	40.2	58.6	15.7	12.8	26.9	33.5	15.3	15.6	17.4	33.3
	Llama3.1-70B	36.0	30.1	44.0	51.8	19.6	12.9	28.4	26.9	15.1	15.4	17.9	46.8
	Qwen2.5-72B	33.1	33.5	76.8	85.7	15.4	16.8	65.8	51.4	14.3	19.6	38.3	53.5
xCOMET	Gemma2-27B	86.0	96.1	68.1	71.8	63.2	77.6	36.3	44.1	45.2	46.7	27.0	25.0
	Llama3.1-70B	86.8	95.6	66.1	74.0	63.7	77.4	37.1	46.8	43.5	46.3	27.8	28.9
	Qwen2.5-72B	87.6	95.6	24.5	30.1	65.2	76.3	20.3	28.4	45.0	45.7	18.0	18.4
Retention Rate	Gemma2-27B	1.00	1.08	0.98	0.99	0.98	0.98	1.07	0.81	1.02	0.96	0.89	0.95
	Llama3.1-70B	1.01	1.06	1.00	0.97	0.92	1.06	0.99	0.77	1.01	0.98	0.91	0.89
	Qwen2.5-72B	1.03	1.04	0.71	0.71	0.90	0.98	1.04	0.79	1.00	1.04	0.93	0.86

Table 6: There exists challenges in domain-specific translation evaluation. The table presents different metric scores of the En-X translation of selected models on specific domains.

Model	Rule-based	Func Compl.	Science
GPT-4o-mini	79.1	78.7	34.1
GPT-4o	80.8	80.1	45.6
DeepSeek-V3	**83.9**	**83.2**	**47.4**

Table 7: The leading open-source model, DeepSeek-V3, bridges the gap to the closed-source models. We compare DeepSeek-V3 and GPT-4o on some of tasks.

that models achieve higher scores generally on our translated tasks compared to naive machine-translated ones. Google Translate lacks the flexibility to handle diverse constraints and specific domains, while GPT-4o-mini does not always perform translation task, especially on instruction data. This indicates that naive machine translation underestimates LLMs' capabilities, whereas our data provides a more accurate assessment.

5.3 BenchMAX reveals the challenges in domain-specific translation evaluation

Domain-specific texts often contain substantial segments that do not require translation, such as code, leading to inflated spBLEU scores. To address this, we explore alternative metrics: the edit-distance metric TER (Snover et al., 2006), the model-based metric xCOMET-XXL (Guerreiro et al., 2024), and the performance retention rate that compares downstream task performance builded by model self-translations and human translations. Table 6 presents these metric scores across selected tasks and languages. All these common metrics prove unreliable for domain-specific translation evaluation. Both spBLEU and TER yield extreme values in scientific and code data due to large portions of unchanged text, failing to capture the quality of crucial translated segments. The xCOMET metric is inconsistent, with scores ranging widely (18 to 96), particularly struggling with low-resource languages and specialized domains. Additionally, the performance retention rate provides minimal differentiation between translations, limiting its effectiveness. These findings highlight the need for specialized metrics for domain-specific translation, which is an important direction for future research.

5.4 Comaparing open-source and closed-source models on BenchMAX

As demonstrated in Table 7, although GPT-4o-mini and GPT-4o demonstrates strong multilingual capabilities across various tasks, they fall short of DeepSeek-V3, the leading open-source model in our evaluation. This suggests that state-of-the-art open-source models are becoming competitive with their closed-source counterparts. Due to budget constraints, our evaluation of closed-source models is limited to GPT-4o-mini and GPT-4o on some of tasks. A more comprehensive comparison would be valuable for further validating this trend.

6 Conclusion

We introduce BenchMAX, a comprehensive, high-quality, and parallel multilingual benchmark comprising 10 tasks assessing crucial capabilities across 17 diverse languages. The multilingual task data is initially translated from English using machine translation and subsequently refined through multiple iterations of post-editing by native speakers, ensuring high data quality. Through extensive experiments, we find that the language-agnostic capabilities of current leading LLMs remain uneven across different languages. While increasing model size consistently enhances multilingual performance, the performance gap between English and other languages persists, highlighting the need for further efforts to achieve balanced multilingual capabilities.

7 Limitations

We discuss the limitations of our work in this section.

- The data construction process may introduce model biases, as we use Qwen2.5-72B-Instruct for quality estimation, and GPT-4o-mini for selecting the final translation. However, these biases may have slight influences on the evaluation results, and the overall data quality is high.
- We do not fully evaluate leading multilingual proprietary models such as GPT-4o and Claude-3.7-Sonnet due to the limited resources. Evaluating these models on all tasks, especially the long context task, can cost tens of thousands of dollars.
- In the model-based instruction following task, using LLM-as-a-judge can bring self-bias to the win-rates. This bias stemmed from the nature of LLM-as-a-judge is difficult to circumvent. We provide further analysis of self-bias in Appendix E.

References

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. 2024. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Annual conference of the European Association for Machine Translation*, pages 261–268, Trento, Italy. European Association for Machine Translation.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: An open platform for evaluating llms by human preference. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, et al. 2024. Aya expanse: Combining research breakthroughs for a new multilingual frontier. *arXiv preprint arXiv:2412.04261*.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bing-Li Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dong-Li Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Jun-Mei Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shao-Ping Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wen-Xuan Yu, Wentao Zhang, X. Q. Li, Xiangyu Jin, Xianzu Wang, Xiaoling Bi, Xiaodong Liu, Xiaohan Wang, Xi-Cheng Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yao Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yi Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yi-Bing Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng

Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxiang Ma, Yuting Yan, Yu-Wei Luo, Yu mei You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Zehui Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhen guo Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zi-An Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile van Krieken, and Pasquale Minervini. 2024. Are we done with mmlu?

Gemini. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Nuno M Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André FT Martins. 2024. xcomet: Transparent machine translation evaluation through fine-grained error detection. *Transactions of the Association for Computational Linguistics*, 12:979–995.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

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

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*.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, et al. 2024. Findings of the wmt24 general machine translation shared task: the llm era is here but mt is not solved yet. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1–46.

Viet Lai, Chien Nguyen, Nghia Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan Rossi, and Thien Nguyen. 2023. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. 2024. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2024. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in Neural Information Processing Systems*, 36.

Yinquan Lu, Wenhao Zhu, Lei Li, Yu Qiao, and Fei Yuan. 2024. Llamax: Scaling linguistic horizons of llm by enhancing translation capabilities beyond 100 languages. *arXiv preprint arXiv:2407.05975*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

OpenAI. 2024. Hello gpt-4o.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jirui Qi, Raquel Fernández, and Arianna Bisazza. 2023. Cross-lingual consistency of factual knowledge in multilingual language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, abs/2311.12022.

Angelika Romanou, Negar Foroutan, Anna Sotnikova, Zeming Chen, Sree Harsha Nelaturu, Shivalika Singh, Rishabh Maheshwary, Micol Altomare, Mohamed A Haggag, Alfonso Amayuelas, et al. 2024. Include: Evaluating multilingual language understanding with regional knowledge. *arXiv preprint arXiv:2411.19799*.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2023. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations*.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David I. Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, Raymond Ng, Shayne Longpre, Wei-Yin Ko, Madeline Smith, Antoine Bosselut, Alice Oh, Andre F. T. Martins, Leshem Choshen, Daphne Ippolito, Enzo Ferrante, Marzieh Fadaee, Beyza Ermis, and Sara Hooker. 2024. Global mmlu: Understanding and addressing cultural and linguistic biases in multilingual evaluation.

Matthew G. Snover, Bonnie J. Dorr, Richard M. Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers, AMTA 2006, Cambridge, Massachusetts, USA, August 8-12, 2006*, pages 223–231. Association for Machine Translation in the Americas.

Venkat Krishna Srinivasan, Zhen Dong, Banghua Zhu, Brian Yu, Damon Mosk-Aoyama, Kurt Keutzer, Jiantao Jiao, and Jian Zhang. 2023. Nexusraven: a commercially-permissive language model for function calling. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Qiushi Sun, Zhirui Chen, Fangzhi Xu, Kanzhi Cheng, Chang Ma, Zhangyue Yin, Jianing Wang, Chengcheng Han, Renyu Zhu, Shuai Yuan, et al. 2024. A survey of neural code intelligence: Paradigms, advances and beyond. *arXiv preprint arXiv:2403.14734*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

InternLM Team. 2023. Internlm: A multilingual language model with progressively enhanced capabilities.

Bin Wang, Zhengyuan Liu, Xin Huang, Fangkai Jiao, Yang Ding, AiTi Aw, and Nancy Chen. 2024a. SeaEval for multilingual foundation models: From cross-lingual alignment to cultural reasoning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024b. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Wang. 2024. Pride and prejudice: LLM amplifies self-bias in self-refinement. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15474–15492, Bangkok, Thailand. Association for Computational Linguistics.

Fei Yuan, Linjun Shou, Xuanyu Bai, Ming Gong, Yaobo Liang, Nan Duan, Yan Fu, and Daxin Jiang. 2020. Enhancing answer boundary detection for multilingual machine reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Yidan Zhang, Boyi Deng, Yu Wan, Baosong Yang, Haoran Wei, Fei Huang, Bowen Yu, Junyang Lin, Fei Huang, and Jingren Zhou. 2024. P-mmeval: A parallel multilingual multitask benchmark for consistent evaluation of llms.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024. Multilingual machine translation with large language models: Empirical results and analysis. In *Findings of the Association for Computational Linguistics: NAACL 2024*.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1. 0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534.

A Capability and Task Data Selection

Instruction Following Capability involves understanding and executing commands accurately and efficiently. In the light of varied evaluation methods - rule-based or model-based - we include two distinct tasks.

- Rule-based Intruction Following: We collect data from IFEval (Zhou et al., 2023), which is a benchmark for evaluating the instruction following abilities of LLMs, composed of around 500 verifiable instructions and can be evaluated for accuracy using automated rules. Note that the accuracy for IFEval is the average of the four accuracies (i.e. prompt-strict, prompt-loose, inst-strict and inst-loose accuracies), following (Dubey et al., 2024).
- Model-based Instruction Following: We collect data from Arena-hard (Li et al., 2024) which contains 500 real-world instructions from the Chatbot Arena (Chiang et al., 2024), and m-ArenaHard[3] which contains translated multilingual versions. This benchmark can provide better model separability and higher alignment with human preference. It is assessed by the Win Rate of the testing model in comparison to the baseline model, GPT-4o, judged against DeepSeek-V3.

Code Generation Capability refers to automatically producing functional code scripts based on given requirements. Considering variations in difficulty, two separate tasks are included.

- Function Completion: We collect data from Humaneval+ (Liu et al., 2024) which is an augmented version of HumanEval (Chen et al., 2021), comprising an expanded test cases. Each problem in the benchmark gives a definition of a Python function accompanied by an English docstring, and requires LLMs to complete the function.
- Problem Solving: We collect data from LiveCodeBench [4] (Jain et al., 2024) which provides a more rigorous assessment of the code generation capabilities. It is a much harder benchmark by collecting coding problems in natural language from real competition platforms with live updates.

Long Context Modeling Capability involves understanding and generating coherent text from extensive input sequences, allowing the model to capture dependencies and relationships within lengthy texts. This paper focuses on the long-context evaluation of multilingual settings based on the RULER benchmark (Hsieh et al., 2024).

- Question Answering: We build synthetic testsets based on RULER, which contains several question answering long-context tasks with predefined context length, such as the needle-in-a-haystack (NIAH) test and question answering (QA) test. Since the NIAH test is unrealistic and many models perform perfectly on it, we add a new task called QA-in-a-heystack (QAIAH), where one or several paragraphs are inserted into the haystack. The model then answers the question related to the inserted paragraph instead of finding the obtrusive needle. We reserve the tasks of NIAH, QAIAH, and variable tracking (VT) in our task list, while others are excluded.

Reasoning encompasses thinking logically, drawing conclusions, making inferences, and solving problems by processing data, applying rules, and utilizing various forms of logic and knowledge representation. Pushing LLMs beyond surface-level tasks, we extend MGSM (Shi et al., 2023) and GPQA (Rein et al., 2023) requiring deeper understanding and reasoning across different context.

- Math Reasoning: We collect data from MGSM which evaluates the capability of LLM to solve math reasoning problems in multiple languages, focusing on grade-school level complexity.
- Science Reasoning: We collect data from GPQA which is crucial for assessing LLM capability for advanced, unsearchable reasoning and critical thinking across diverse, complex domains. It comprises multiple choice questions formulated by experts in the domains of biology, physics, and chemistry, posing extreme challenges where human experts achieve accuracy lower than 70%.

Tool Use Capability requires the model to translate user queries into executable functions for calling in operating software tools. We extend Nexus (Srinivasan et al., 2023) to a multilingual version, which is adopted by Llama3 (Dubey et al., 2024).

- Multiple Functions: Nexus offers a set of functions and user queries. For each query, the language model is required to generate a function call from a list of noisy functions, in accordance with the function definitions and docstrings.

[3] https://huggingface.co/datasets/CohereForAI/m-ArenaHard

[4] We adopt the code generation subset in LiveCodeBench v4 as the original English dataset.

[Original Text] {prompt: Create an ad copy by expanding “Get 40 miles per gallon on the highway” in the form of a QA with a weird style. Your response should contain less than 8 sentences. Do not include keywords ‘mileage’ or ‘fuel’ in your response. instruction_id_list: [‘length_constraints: number_sentences’, ‘keywords: forbidden_words’] kwargs: [{’relation’: ‘less than’, ‘num_sentences’: 8}, {‘forbidden_words’: [‘mileage’, ‘fuel’]}]}
[Translation Input] Create an ad copy by expanding "Get 40 miles per gallon on the highway" in the form of a QA with a weird style. Your response should contain less than 8 sentences. Do not include keywords ‘<b>mileage</b>’ or ‘<b>fuel</b>’ in your response.
[Google Translation Result] 以风格怪异的问答形式扩展“在高速公路上每加仑行驶 40 英里”来创建广告文案。您的回复应少于 8 个句子。请勿在回复中包含关键字“<b>里程</b>”或“<b>燃料</b>”。
[Postprocessing] {prompt: 以风格怪异的问答形式扩展“在高速公路上每加仑行驶 40 英里”来创建广告文案。您的回复应少于 8 个句子。请勿在回复中包含关键字“里程”或“燃料”。 instruction_id_list: [‘length_constraints: number_sentences’, ‘keywords: forbidden_words’] kwargs: [{‘relation’: ‘less than’, ‘num_sentences’: 8}, ‘forbidden_words’: [‘里程’, ‘燃料’]] }
[Human Post-Editing] {prompt: 以一种奇特风格的问答形式展开“在高速公路上每加仑行驶40英里”这句话，创建为一个广告文案。你的回答应该少于8句话。不要在你的回复中包含关键字“里程”或“燃料”。 instruction_id_list: [‘length_constraints: number_sentences’, ‘keywords: forbidden_words’] kwargs: [{‘relation’: ‘less than’, ‘num_sentences’: 8}, ‘forbidden_words’: [‘里程’, ‘燃料’]] }

Table 8: One example in rule-based instruction following task, which includes complex constraints. First, we enclose these constraints with special symbols and then translate the prompt from English to the target language by Google Translate. Finally, we postprocess the prompt by extracting constraints into kwargs and removing special symbols for human post-editing.

Translation Capability needs the model to convert text between multiple languages while maintaining semantic meaning accurately. To comprehensively evaluate this capability, we introduce general and task-specific translation datasets.

- General: General domain data are composed of Flores-200 (Costa-jussà et al., 2022), TED (Cettolo et al., 2012) and WMT24 (Kocmi et al., 2024) testsets. In BenchMAX, we include parallel data from 17 selected languages.
- Domain: Domain translation data is a by-product of the BenchMAX construction process, encompassing a 17-way parallel task across diverse domains, such as reasoning, code generation, tool usage, and instruction following. Unlike traditional translation tasks, this poses a new challenge to the model by requiring it to determine whether a given segment should be translated or not.

B Dataset Construction

B.1 Rule-based Instruction Following Dataset

We first filter out some English-specific instructions from the original dataset, such as changing the letter cases. After filtering, the number of remaining samples is 429. The next problem is how to extract the keywords from the translated instruction since the keywords are also translated and are required in the verification step.

For example, as shown in Table 8, the sample requires extra processing to extract constraints from the translated instruction, as they are needed for verification. Inspired by Yuan et al. (2020), we enclose the keywords in the original instruction with special symbols, making them easy to extract from the translated result. If one symbol fails, another symbol is used to improve recall.

Setting		Target Language zh	es	fr	hu
w/o special symbols		0.68	0.68	0.68	0.68
symbol 1: <b> </b>		0.91	0.89	0.88	0.93
symbol 2: ( )		0.88	0.91	0.89	0.92
symbol 3: ([ ])		0.82	0.89	0.87	0.92
Order 1	+ symbol 1	0.91	0.89	0.88	0.93
	+ symbol 2	0.93	0.93	0.90	0.95
Order 2	+ symbol 2	0.88	0.91	0.89	0.92
	+ symbol 1	0.90	0.93	0.90	0.95
	+ symbol 3	0.92	0.93	0.90	0.95

Table 9: The recall rates of constraints using different groups of special symbols. We choose Order 1, which has fewer steps and produces on-par or better performance than other settings.

As shown in Table 9, we explore various groups of special symbols and different orders, and calculate the recall rates of keywords. Comparing to not using special symbols, apply any symbol group can greatly improve the recalls, while combining different symbol groups in multiple rounds can further improve the recalls. We choose *Or-*

	zh	es	fr	de	hu	ru	ja	th	sw	bn	te	ar	ko	vi	cs	sr
w/o special symbols	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68
+symbol 1	0.91	0.89	0.88	0.92	0.93	0.93	0.91	0.91	0.92	0.95	0.96	0.90	0.88	0.90	0.92	0.99
+symbol 2	0.93	0.93	0.90	0.92	0.95	0.94	0.92	0.93	0.93	0.97	0.99	0.94	0.91	0.92	0.93	1.00

Table 10: The recall of keywords when translating IFEval English data to other languages.

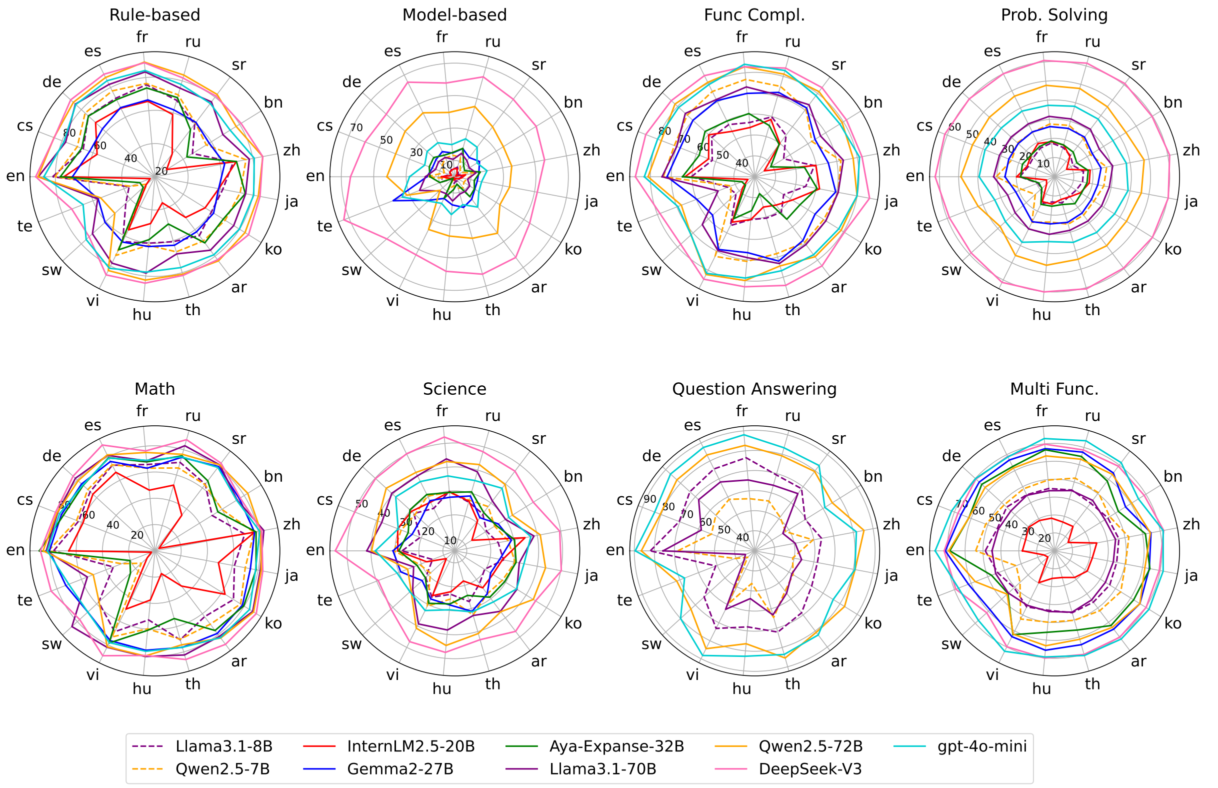


Figure 7: The radar charts visualize the performance of models on each subtask in different languages. Most model evaluated have imbalanced performance across different languages.

der 1 as it can achieve better results with fewer groups than *Order 2*. Complete results across all languages are provided in Table 10. In addition, the number-word constraints for non-English languages are multiplied by a ratio in order to make the difficulty of the same instruction in different languages comparable. Specifically, we calculate the ratio of the word count of English to that of a non-English language in the Flores-200 corpus using language-specific tokenizers. we also adapt verification rules to multilingual scenarios. For instance, word and sentence segmentation methods may vary across different languages.

During post-editing, we ask human annotators to check whether the translated keywords in the kwargs, which are used by the rule-based program, appear in the translated instruction.

B.2 Model-based Instruction Following Dataset

Ten of the sixteen languages required have been provided by m-ArenaHard, which has translated the original dataset into 22 languages using Google Translate. Based on m-ArenaHard, we further translate the English data into six other languages via Google Translate. Subsequently, we ask human annotators to review and edit the translated instructions in all 16 languages.

B.3 Function Completion Dataset

The objective is to translate only the natural texts within the function comments. However, it is challenging to prevent Google Translate from translating other elements, such as function names. Alternatively, we instruct GPT-4o to complete this translation task with well-designed prompts (Table 17). Furthermore, a human post-editing process is employed to refine the quality of the generated translation.

B.4 Problem Solving Dataset

Similar to the Function Completion Dataset, we employ GPT-4o to translate the English problems into other 16 languages with a well-designed prompt (Table 18), since Google Translate cannot distinguish the parts that should remain untranslated. Human review is also used to ensure the overall quality of the translated texts.

B.5 Math Reasoning Dataset

Given that the MGSM examples are written in ten languages we need, we only translate the English version into the remaining six languages via Google Translate. This is also followed by a manual checking procedure.

B.6 Science Reasoning Dataset

The question and the four options of each sample are translated into 16 other languages by Google Translate. In particular, the question and options are concatenated by option markers like "(A)". After translation, we extract the translated question and options to form a new sample.

B.7 Long-Context Question Answering Dataset

The haystacks, needles, paragraphs and questions related to QAs are translated to other languages. We use the parallel testsets from the UN corpus (Ziemski et al., 2016) as the haystack. The English version contains about 128k tokens, and we extend it to other languages using Google Translate. The sentence of the needle is also translated into 16 other languages, in which UUIDs are employed as keys and values that are not translated. With respect to the QA data, we translate the paragraphs and questions in XQuAD (Artetxe et al., 2020) to the languages we need. Note that we also use the trick in translating IFEval to extract the answer spans. With access to our multilingual haystacks, needles and paragraphs, we are able to synthesize the multilingual long-context testsets.

B.8 Multiple Functions Dataset

We only translate the user queries from English into other languages, given that the majority tool descriptions are written in English. The user queries are initially translated by Google Translate and subsequently adjusted by human annotators. To preserve the English parameters, we replace them with placeholders before machine translation and restore them afterward.

C Model Information

Here we list the evaluated models in this section.

Llama3.1-Instruct (Dubey et al., 2024) series contains three multilingual large language models with number of parameters ranging from 8B to 405B. The pre-training corpus of Llama3.1 contains 8% multilingual tokens, and multilingual alignment is also optimized during post-training. In our experiments, we evaluate the 8B version and the 70B version of Llama3.1-Instruct.

Qwen2.5-Instruct (Qwen Team, 2024) is a collection of multilingual language models with several sizes, ranging from 0.5B to 72B. The models are trained with multilingual tokens in both pre-training stage and post-training stage, and are rigorously evaluated on several multilingual tasks. In our experiments, we evaluate the 7B, 32B ,and 72B version of Qwen2.5-Instruct.

Aya-Expanse (Dang et al., 2024) is an open-weight research of models with advanced multilingual capabilities, supporting 23 languages. The Aya Expanse 8B and 32B variants are instruction-tuned and beat Llama3.1-instruct models on the m-ArenaHard, a multilingual instruction following benchmark.

Gemma2-IT (Team et al., 2024) family demonstrates strong multilingual capabilities, although this is not highlighted in the technical report. We benchmark the 9B and 27B variants of Gemma2-IT.

InternLM2.5-chat (Cai et al., 2024) is the successor of InternLM (Team, 2023), which is claimed as a multilingual model. We include the 7B version and 20B version in our experiments. InternLM2.5-7B-chat-1m is a long-context variant supporting context windows with 1M tokens.

DeepSeek-V3 (DeepSeek-AI et al., 2024) is one of state-of-the-art open-source models that achieve performance comparable to that of the best proprietary models. It is a 671B MoE model, with 37B activated for each token. A multilingual corpus and a multilingual-optimized tokenizer are incorporated into their training process.

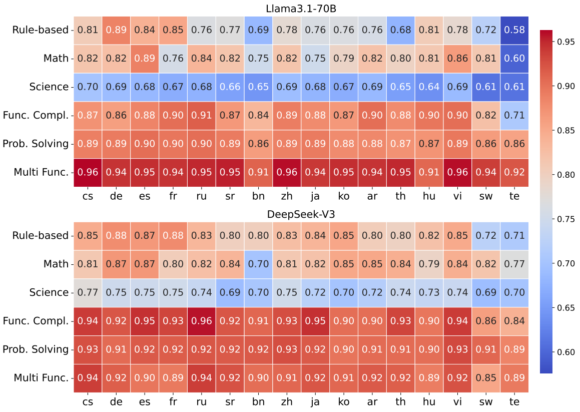


Figure 8: Advanced models show high consistency between English and other languages across six tasks.

DeepSeek-R1-Distill-Llama & DeepSeek-R1-Distill-Qwen (Guo et al., 2025) are dense models with long reasoning capabilities, and are distilled from DeepSeek-R1 based on Llama3.1-8B, Llama3.3-70B-Instruct, Qwen2.5-Math-7B, and Qwen2.5-32B.

GPT-4o & GPT-4o-mini (OpenAI, 2024) are two of the best proprietary models that also achieve remarkable performance on multilingual tasks. Their tokenizer can better compress multilingual texts than that of GPT-4. GPT-4o-mini is the smaller version of GPT-4o with powerful performance.

D Inference Configuration

We adopt greedy decoding for most tasks, except for the problem solving task, where the sampling temperature is set to 0.2. The default chat template and system prompt of each model are applied. Detailed prompts are provided in Appendix G. For reasoning tasks, we adopt the zero-shot native chain-of-thought templates in LM-Evaluation-Harness (Gao et al., 2024). For other tasks, we use the prompt templates provided in corresponding repositories[5], and change the user inputs to other languages.

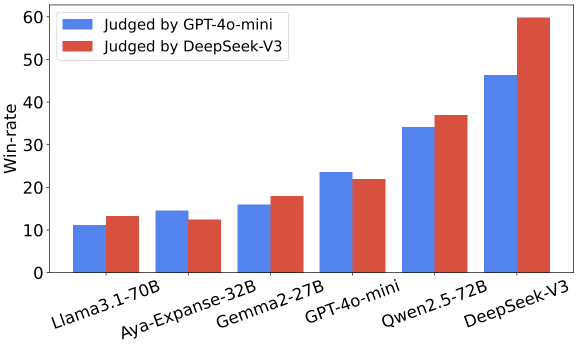


Figure 9: Self-bias is inevitable in model-based instruction following evaluation. DeepSeek-V3 prefers its own outputs, while GPT-4o-mini tends to prefer GPT-4o's outputs. The win-rates of evaluated models are judged by DeepSeek-V3 and GPT-4o-mini.

E More Analysis

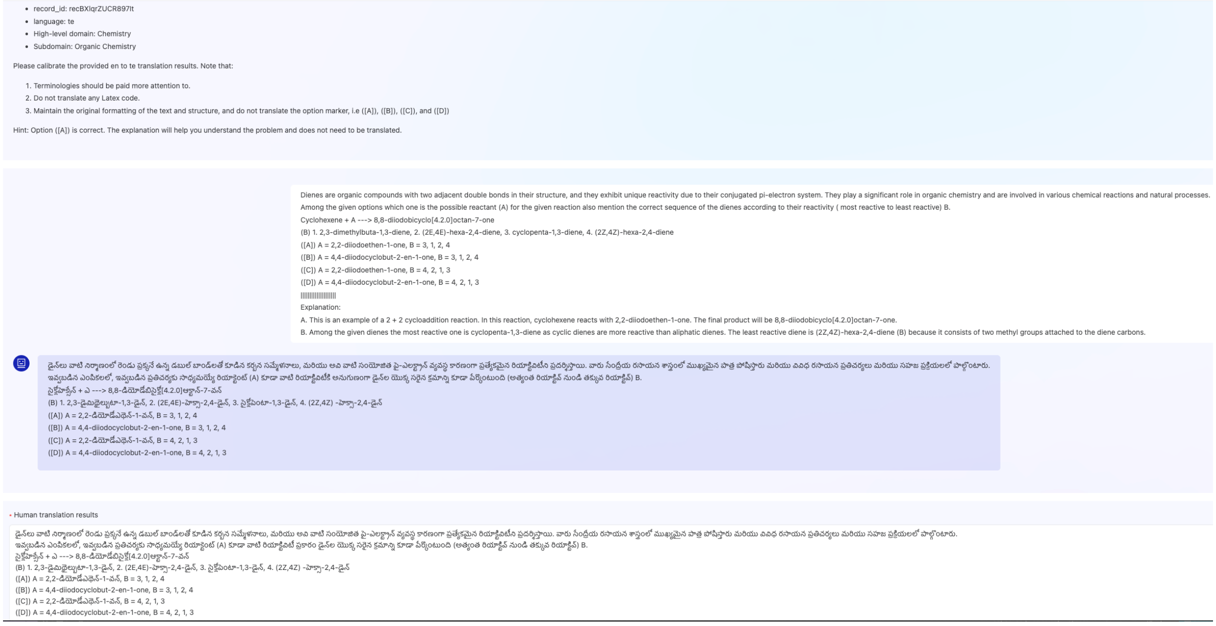
E.1 High consistency between the questions answered correctly/incorrectly in English and in other languages

Although sometimes similar performance can be achieved across different languages for certain tasks, the specific problems being addressed may vary significantly. To examine the language alignment, we compute the consistency between the problem-solving correctness in English versus other languages. Consistency is calculated as the proportion of predictions where a model's output is correct or incorrect in both languages, out of all evaluated samples. Figure 8 presents the consistency between English and languages, based on results of Llama3.1-70B and DeepSeek-V3 on six subtasks of BenchMAX. Both these strong multilingual models demonstrate high consistency on most tasks, with most scores exceeding 0.75. Agreement for low-resource languages are notably lower than those for high-resource languages. Low consistency is also pronounced in science reasoning tasks, suggesting these knowledge-intensive problems pose unique challenges for cross-lingual knowledge transfer.

E.2 Self-bias is inevitable in the model-based instruction following tasks

Applying model-based evaluation exhibits self-bias, where the judge model prefers the outputs of it-

[5] https://github.com/EleutherAI/lm-evaluation-harness
https://github.com/LiveCodeBench/LiveCodeBench
https://github.com/evalplus/evalplus
https://github.com/NVIDIA/RULER
https://github.com/lmarena/arena-hard-auto



- record_id: recBXlqrZUCR897lt
- language: te
- High-level domain: Chemistry
- Subdomain: Organic Chemistry

Please calibrate the provided en to te translation results. Note that:

1. Terminologies should be paid more attention to.
2. Do not translate any Latex code.
3. Maintain the original formatting of the text and structure, and do not translate the option marker, i.e ([A]), ([B]), ([C]), and ([D])

Hint: Option ([A]) is correct. The explanation will help you understand the problem and does not need to be translated.

Dienes are organic compounds with two adjacent double bonds in their structure, and they exhibit unique reactivity due to their conjugated pi-electron system. They play a significant role in organic chemistry and are involved in various chemical reactions and natural processes.
Among the given options which one is the possible reactant (A) for the given reaction also mention the correct sequence of the dienes according to their reactivity (most reactive to least reactive) B.
Cyclohexene + A ---> 8,8-diiodobicyclo[4.2.0]octan-7-one
(B) 1. 2,3-dimethylbuta-1,3-diene, 2. (2E,4E)-hexa-2,4-diene, 3. cyclopenta-1,3-diene, 4. (2Z,4Z)-hexa-2,4-diene
([A]) A = 2,2-diiodoethen-1-one, B = 3, 1, 2, 4
([B]) A = 4,4-diiodocyclobut-2-en-1-one, B = 3, 1, 2, 4
([C]) A = 2,2-diiodoethen-1-one, B = 4, 2, 1, 3
([D]) A = 4,4-diiodocyclobut-2-en-1-one, B = 4, 2, 1, 3
||||||||||||||||||
Explanation:
A. This is an example of a 2 + 2 cycloaddition reaction. In this reaction, cyclohexene reacts with 2,2-diiodoethen-1-one. The final product will be 8,8-diiodobicyclo[4.2.0]octan-7-one.
B. Among the given dienes the most reactive one is cyclopenta-1,3-diene as cyclic dienes are more reactive than aliphatic dienes. The least reactive diene is (2Z,4Z)-hexa-2,4-diene (B) because it consists of two methyl groups attached to the diene carbons.

డైన్‌లు వాటి నిర్మాణంలో రెండు ప్రక్కనే ఉన్న డబుల్ బాండ్‌లతో కూడిన కర్బన సమ్మేళనాలు, మరియు అవి వాటి సంయోజిత పై-ఎలక్ట్రాన్ వ్యవస్థ కారణంగా ప్రత్యేకమైన రియాక్టివిటీని ప్రదర్శిస్తాయి. వారు సేంద్రియ రసాయన శాస్త్రంలో ముఖ్యమైన పాత్ర పోషిస్తారు మరియు వివిధ రసాయన ప్రతిచర్యలు మరియు సహజ ప్రక్రియలలో పాల్గొంటారు.
ఇవ్వబడిన ఎంపికలలో, ఇవ్వబడిన ప్రతిచర్యకు సాధ్యమయ్యే రియాక్టెంట్ (A) కూడా వాటి రియాక్టివిటీకి అనుగుణంగా డైన్‌ల యొక్క సరైన క్రమాన్ని కూడా పేర్కొంటుంది (అత్యంత రియాక్టివ్ నుండి తక్కువ రియాక్టివ్) B.
సైక్లోహెక్సేన్ + ఎ ---> 8,8-డియోడోబిసైక్లో[4.2.0]ఆక్టాన్-7-వన్
(B) 1. 2,3-డైమిథైల్బుటా-1,3-డైన్, 2. (2E,4E)-హెక్సా-2,4-డైన్, 3. సైక్లోపెంటా-1,3-డైన్, 4. (2Z,4Z) -హెక్సా-2,4-డైన్
([A]) A = 2,2-డియోడోఎథెన్-1-వన్, B = 3, 1, 2, 4
([B]) A = 4,4-diiodocyclobut-2-en-1-one, B = 3, 1, 2, 4
([C]) A = 2,2-డియోడోఎథెన్-1-వన్, B = 4, 2, 1, 3
([D]) A = 4,4-diiodocyclobut-2-en-1-one, B = 4, 2, 1, 3

* Human translation results

డైన్‌లు వాటి నిర్మాణంలో రెండు ప్రక్కనే ఉన్న డబుల్ బాండ్‌లతో కూడిన కర్బన సమ్మేళనాలు, మరియు అవి వాటి సంయోజిత పై-ఎలక్ట్రాన్ వ్యవస్థ కారణంగా ప్రత్యేకమైన రియాక్టివిటీని ప్రదర్శిస్తాయి. వారు సేంద్రియ రసాయన శాస్త్రంలో ముఖ్యమైన పాత్ర పోషిస్తారు మరియు వివిధ రసాయన ప్రతిచర్యలు మరియు సహజ ప్రక్రియలలో పాల్గొంటారు.
ఇవ్వబడిన ఎంపికలలో, ఇవ్వబడిన ప్రతిచర్యకు సాధ్యమయ్యే రియాక్టెంట్ (A) కూడా వాటి రియాక్టివిటీ ప్రకారం డైన్‌ల యొక్క సరైన క్రమాన్ని కూడా పేర్కొంటుంది (అత్యంత రియాక్టివ్ నుండి తక్కువ రియాక్టివ్) B.
సైక్లోహెక్సేన్ + ఎ ---> 8,8-డియోడోబిసైక్లో[4.2.0]ఆక్టాన్-7-వన్
(B) 1. 2,3-డైమిథైల్బుటా-1,3-డైన్, 2. (2E,4E)-హెక్సా-2,4-డైన్, 3. సైక్లోపెంటా-1,3-డైన్, 4. (2Z,4Z) -హెక్సా-2,4-డైన్
([A]) A = 2,2-డియోడోఎథెన్-1-వన్, B = 3, 1, 2, 4
([B]) A = 4,4-diiodocyclobut-2-en-1-one, B = 3, 1, 2, 4
([C]) A = 2,2-డియోడోఎథెన్-1-వన్, B = 4, 2, 1, 3
([D]) A = 4,4-diiodocyclobut-2-en-1-one, B = 4, 2, 1, 3

Figure 10: A screenshot of the annotation platform.

self or models from the same family (Li et al., 2024; Xu et al., 2024). We further adopt GPT-4o-mini as the judge model, and compute the win-rate against the baseline model GPT-4o. Figure 9 shows that DeepSeek-V3 strongly favors its own outputs, while GPT-4o-mini prefers GPT-4o's outputs. Nevertheless, the win-rates of other models judged by the different judges are comparable, and the rankings are fairly consistent.

F Details about human annotation

We recuite native annatators to post-edit the machine translations. The instructions vary according to the task and constraints. One screenshot of the annotation platform is shown in Figure 10. The annotators are paid above local average salary.

G Details about Prompt Templates

We present the prompt templates used in each task in this section. Table 11 and Table 12 show the native-CoT prompts for MGSM and GPQA. Table 13 shows the prompt templates for some tasks where the original English template is used. Table 14 shows the prompt templates of the long-context modelling task. Table 16 shows the LLM-Judge Instruction for comparing two translations.

H Detailed results

Figure 7 illustrates the detailed results of each model on each task.

Language	Prompt
En	Question: {question}\nStep-by-Step Answer:
Zh	问题: {question}\n逐步解答:
Es	Pregunta: {question}\nRespuesta paso a paso:
Fr	Question : {question}\nRéponse étape par étape :
De	Frage: {question}\nSchritt-für-Schritt-Antwort:
Ru	Задача: {question}\nпошаговоерешение:
Ja	問題: {question}\nステップごとの答え:
Th	โจทย์: {question}\nคำตอบทีละขั้นตอน:
Sw	Swali: {question}\nJibu la Hatua kwa Hatua:
Bn	প্রশ্ন: {question}\nধাপে ধাপে উত্তর:
Te	ప్రశ్న: {question}\nదశలవారీగా సమాధానం:
Ar	السؤال: {question}\nالإجابة خطوة بخطوة:
Ko	질문: {question}\n단계별 답변:
Vi	Câu hỏi: {question}\nCâu trả lời từng bước:
Cs	Otázka: {question}\nOdpověď krok za krokem:
Hu	Kérdés: {question}\nVálasz lépésről lépésre:
Sr	Питање: {question}\nОдговор корак по корак:

Table 11: The native-CoT prompts of the mathematical reasoning task.

Language	Prompt
En	What is the correct answer to this question:{question}\nChoices:\n(A) {choice1}\n(B) {choice2}\n(C) {choice3}\n(D) {choice4}\nLet's think step by step:
Zh	这个问题的正确答案是什么: {question}\n选项:\n(A) {choice1}\n(B) {choice2}\n(C) {choice3}\n(D) {choice4}\n我们来一步步思考一下:
Es	¿Cuál es la respuesta correcta a esta pregunta? {question}}\nOpciones:\n(A) {choice1}\n(B) {choice2}\n(C) {choice3}\n(D) {choice4}\nPensemos paso a paso:
Fr	Quelle est la bonne réponse à cette question : {question}\nChoix :\n(A) {choice1}\n(B) {choice2}\n(C) {choice3}\n(D) {choice4}\nRéfléchissons étape par étape :
De	Was ist die richtige Antwort auf diese Frage: {question}\nAuswahlmöglichkeiten:\n(A) {choice1}\n(B) {choice2}\n(C) {choice3}\n(D) {choice4}\nLassen Sie uns Schritt für Schritt überlegen:
Ru	Какой правильный ответ на этот вопрос: {question}\nВарианты:\n(A) {choice1}\n(B) {choice2}\n(C) {choice3}\n(D) {choice4}\nДавайте подумаем шаг за шагом:
Ja	この質問の正しい答えは何ですか: {question}\n選択肢:\n(A) {choice1}\n(B) {choice2}\n(C) {choice3}\n(D) {choice4}\nステップごとに考えてみましょう:
Th	คำตอบที่ถูกต้องสำหรับคำถามนี้คืออะไร: {question}\nตัวเลือก:\n(A) {choice1}\n(B) {choice2}\n(C) {choice3}\n(D) {choice4}\nมาคิดทีละขั้นตอนกัน:
Sw	Je, ni jibu gani sahihi kwa swali hili: {question}\nChaguo: \n(A) {choice1}\n(B) {choice2}\n(C) {choice3}\n(D) {choice4}\nWacha tufikirie hatua kwa hatua:
Bn	এই প্রশ্নের সঠিক উত্তর কি: {question}\nপছন্দ:\n(A) {choice1}\n(B) {choice2}\n(C) {choice3}\n(D) {choice4}\nআসুন ধাপে ধাপে চিন্তা করি:
Te	ఈ ప్రశ్నకు సరైన సమాధానం ఏమిటి: {question}\nఎంపికలు:\n(A) {choice1}\n(B) {choice2}\n(C) {choice3}\n(D) {choice4}\nదశలవారీగా ఆలోచిద్దాం:
Ar	ما هي الإجابة الصحيحة لهذا السؤال: {question}\nالخيارات:\n(A) {choice1}\n(B) {choice2}\n(C) {choice3}\n(D) {choice4}\nلنفكر خطوة بخطوة:
Ko	이 질문에 대한 정답은 무엇입니까? {question}\n선택지:\n(A) {choice1}\n(B) {choice2}\n(C) {choice3}\n(D) {choice4}\n단계별로 생각해 보겠습니다:
Vi	Câu trả lời đúng cho câu hỏi này là gì: {question}\nCác lựa chọn:\n(A) {choice1}\n(B) {choice2}\n(C) {choice3}\n(D) {choice4}\nChúng ta hãy suy nghĩ từng bước một:
Cs	Jaká je správná odpověď na tuto otázku: {question}\nMožnosti:\n(A) {choice1}\n(B) {choice2}\n(C) {choice3}\n(D) {choice4}\nZamysleme se krok za krokem:
Hu	Mi a helyes válasz erre a kérdésre: {question}\nVálasztási lehetőségek:\n(A) {choice1}\n(B) {choice2}\n(C) {choice3}\n(D) {choice4}\nGondoljuk végig lépésről lépésre:
Sr	Који је тачан одговор на ово питање: {question}\nИзбори:\n(A) {choice1}\n(B) {choice2}\n(C) {choice3}\n(D) {choice4}\nХајде да размислимо корак по корак:

Table 12: The native-CoT prompts of the scientific reasoning task.

Task	Prompt Template
Rule-based instruction following	{prompt}
Model-based instruction following	{prompt}

Problem Solving	**[System Message]**
	You are an expert Python programmer. You will be given a question (problem specification) and will generate a correct Python program that matches the specification and passes all tests. You will NOT return anything except for the program.
	[User Message]
	### Question:
	{question}
	### Format: Read the inputs from stdin solve the problem and write the answer to stdout (do not directly test on the sample inputs). Enclose your code within delimiters as follows.
	```python
	# YOUR CODE HERE
	```
	
	### Answer: (use the provided format with backticks)

Function Completion	**[User Message]**
	Please provide a self-contained Python script that solves the following problem in a markdown code block:
	```
	{prompt}
	```
	[Assistant Message]
	Below is a Python script with a self-contained function that solves the problem and passes corresponding tests:
	``python

Tool use	**[Tool Info]** {prompt}

Table 13: The prompt templates of the listed tasks. The prompt in the template is multilingual.

Subtask	Prompt Template
NIAH	**[User Message]** Some special magic uuids are hidden within the following text. Make sure to memorize it. I will quiz you about the uuids afterwards. {heystack} What are all the special magic uuids for {query} mentioned in the provided text? **[Assistant Message]** The special magic uuids for {query} mentioned in the provided text are
QA in a heystack (QAIAH)	**[User Message]** Answer the questions based on the given documents. Only give me the answers and do not output any other words. The following are given documents. {context} Answer the questions based on the given documents. Only give me the answers and do not output any other words. Questions: {query} **[Assistant Message]** Answers:
Variable Tracking (VT)	**[User Message]** Memorize and track the chain(s) of variable assignment hidden in the following text. {context} Question: Find all variables that are assigned the value {query} in the text above. **[Assistant Message]** Answer: According to the chain(s) of variable assignment in the text above, 5 variables are assgined the value {query}, they are:
QA	**[User Message]** Answer the question based on the given documents. Only give me the answer and do not output any other words. The following are given documents. {context} Answer the question based on the given documents. Only give me the answer and do not output any other words. Question: {query} **[Assistant Message]** Answer:

Table 14: The prompt templates of the long-context modelling task.

Score the following translation from {src_lang} to {tgt_lang} with respect to the human reference on a continuous scale from 0 to 100 that starts with “No meaning preserved”, goes through “Some meaning preserved”, then “Most meaning preserved and few grammar mistakes”, up to “Perfect meaning and grammar”

{src_lang} source: “{source}”
{tgt_lang} translation: “{target}”
Score:

Table 15: The GEMBA-SQM prompt.

[System Message]
Please act as an impartial judge and evaluate the quality of the lang translations provided by two humans for the English source sentence displayed below. You will be given human A’s translation and human B’s translation. Your job is to evaluate which human’s translation is better.

You must identify and correct any mistakes or inaccurate information.

Consider if the human’s translations are accurate and fluent. Accurate means the translation conveys the same meaning, information, and nuances as the original source text. Fluent refers to the quality of the translation in terms of its naturalness, readability, and adherence to the grammatical, stylistic, and idiomatic conventions of the target language.

Then consider whether the human’s translations are consistent with the context. Code input/output and programming language syntax should not be translated. Finally, review the formatting of the translated text, including indentation, to ensure it is consistent and appropriate.

After providing your explanation, you must output only one of the following choices as your final verdict with a label:

1. Human A is significantly better: [[A>>B]]
2. Human A is slightly better: [[A>B]]
3. Tie, relatively the same: [[A=B]]
4. Human B is slightly better: [[B>A]]
5. Human B is significantly better: [[B>>A]]

Example output: “My final verdict is tie: [[A=B]]”.
[User Message]
<| Source Text|>
{source}

<|The Start of Human A’s Translation|>
{translation_1}
<|The End of Human A’s Translation|>

<|The Start of Human B’s Translation|>
{translation_2}
<|The End of Human B’s Translation|>

Table 16: LLM-Judge Instruction

[System Message]
You are a professional translator specializing in technical content. Please translate the following English Python codes into {tgt_lang}, adhering to these specific guidelines:

1. **Do not translate** content representing code input/output or programming language syntax. Only translate content in comments.
2. **Maintain the original formatting** of the text, structure and indentation.
3. **Do not translate** any LaTeX code.
4. **Only output the translation** without any additional comments or explanations.
[User Message]
{problem}

Table 17: Prompt for translating the Function Completion task.

[System Message]
You are a professional translator specializing in technical content. Please translate the following English coding problems into {tgt_lang}, adhering to these specific guidelines:

1. **Do not translate** any LaTeX code.
2. **Do not translate** content representing code input/output or programming language syntax.
3. **Maintain the original formatting** of the text and structure.
4. **Only output the translation** without any additional comments or explanations.
[User Message]
{problem}

Table 18: Prompt for translating the Problem Solving task.