Which Language Should a Skeleton Speak? Language Choices in Multilingual Reasoning

HyeonSeok Lim[1], **SeungWoo Song**[2], **Inho Won**[2], **Hoyun Song**[2,3], **Jihyo Kim**[2]
KyungTae Lim[2†]

[1]ETRI [2]KAIST [3]Department of Artificial Intelligence, Dankook University
gustjrantk@etri.re.kr, {sswoo,kotmul2no,jihyo.kim,ktlim}@kaist.ac.kr,
hoyun.song@dankook.ac.kr

Abstract

Skeleton-based reasoning prompting is a promising training-free approach for structuring LLM reasoning, but prior work largely assumes an English-centric setting. We propose the Language-Aware Skeleton Exploration Framework (LASEF) to study skeleton-language choice in multilingual mathematical reasoning. Across math benchmarks, model scales, and languages, we show that English skeletons yield a small positive tendency on average, most visible for smaller models and low-resource languages. However, few language-level gains remain significant after correction, and English is not universally optimal. Combining greedy decoding, multi-rollout evaluation, translation ablation, and cross-benchmark validation, we further find three patterns of skeleton-language effects: directionally consistent, evaluation- and benchmark-dependent, and asymmetric negative. These effects cannot be fully explained by generation quality alone. Overall, skeleton language is a context-dependent design variable that requires multi-level exploration. All resources are released[1].

1 Introduction

Recent studies on structured reasoning prompting have employed an approach (Wang et al., 2023; Khot et al., 2022). This approach guides a model's reasoning process by concisely presenting the core flow of thought required for problem solving. In this paper, we refer to such explicit representations of reasoning flow as a *skeleton*. A skeleton provides a high-level indication of the direction or strategy for solving a given input query, upon which more detailed reasoning and the final answer are subsequently generated (Ning et al., 2023; Li et al., 2024; Qi et al., 2025c). A key property of skeleton-based reasoning is its *training-free* nature, which enables its use in low-resource language scenarios.

However, when applying skeletons in multilingual settings, there exists a fundamental design challenge. A skeleton-based reasoning pipeline implicitly involves three language choices: (1) the language of the input query, (2) the language in which the skeleton is expressed, and (3) the language in which reasoning and the final answer are generated. Despite the fact that different combinations of these elements may lead the model to rely on different linguistic priors and reasoning trajectories (Wang et al., 2025a, 2024), prior work has not systematically explored this design space of language combinations.

Prior work on skeleton-based reasoning has largely assumed an English-centric setting (Zhou et al., 2022; Qi et al., 2025c), focusing on how to design the structure or prompt format of skeletons to improve performance. While some studies have applied skeletons to multilingual or cross-domain scenarios (Qin et al., 2023; Huang et al., 2023), most of them either fix the skeleton language to English or treat language choice as a secondary factor. As a result, the crucial question of which language should be utilized for the reasoning skeleton remains insufficiently analyzed for real-world multilingual deployment.

Meanwhile, recent prior studies have reported that even the same model can exhibit different performance depending on the language in which reasoning is carried out (Qi et al., 2025b), and have shown that performing reasoning after translating the query into English can be effective (Shi et al., 2022; Liu et al., 2025; Zhu et al., 2024). These findings suggest that skeletons are not merely formal tools, but may function as a medium that activates language-specific reasoning priors formed during pretraining (Etxaniz et al., 2024). At the same time, they also imply that the effectiveness of skeletons may not be uniform across languages (Qian et al., 2024; Chen et al., 2024).

In this context, we pose the following research

†Corresponding author.
[1]https://github.com/lhsstn/LASEF

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 31933–31957
October 24-29, 2026 ©2026 Association for Computational Linguistics

question: *Under which linguistic conditions does skeleton-based reasoning operate effectively, and how do different combinations of input language, skeleton language, and reasoning/answer language affect reasoning performance? In particular, how do these effects differ in low-resource languages?* This question goes beyond asking "which skeleton is effective," and instead addresses a more fundamental design issue: which language should be used for the skeleton.

To investigate this question, we propose *Language-Aware Skeleton Exploration*. Here, exploration does not involve any new algorithmic learning or parameter updates; instead, it refers to an empirical approach that systematically analyzes performance variations across different language combinations while keeping the skeleton structure and format fixed, with the goal of identifying effective language configurations. This setup allows us to disentangle the impact of skeleton design itself from that of language choice.

Our study focuses on mathematical reasoning, a domain where multilingual performance gaps are known to be significant due to limited reasoning capabilities in non-English languages (Qi et al., 2025b). We conduct experiments across benchmarks of varying difficulty levels, models of different scales, and a wide range of language settings spanning high-resource to low-resource languages. Through this comprehensive evaluation, we systematically analyze how the effectiveness of skeletons varies across languages and conditions. Building on these findings, we derive practical insights for the reliable use of skeleton-based reasoning in multilingual environments. Our results answer the above question conditionally rather than universally: instead of prescribing a single optimal skeleton language, we characterize when skeleton-language choices help or hurt and how consistently these effects generalize across decoding protocols and benchmarks. In summary, the contributions of this paper are as follows:

- We formalize the combination of input, skeleton, and reasoning languages as a core design variable in multilingual skeleton-based reasoning, and propose a training-free framework (LASEF) to analyze this design space.
- Combining greedy decoding and multi-rollout evaluation across math benchmarks of varying difficulty, we demonstrate the non-uniform effectiveness of skeleton languages and assess the directional consistency of the observed effects across decoding protocols and benchmarks.
- We identify three distinct patterns of skeleton-language effects: directionally consistent, evaluation- and benchmark-dependent, and asymmetric negative. These patterns are supported by a translation ablation that disentangles skeleton generation quality from cross-lingual alignment.

2 Related Work

English-Pivot Reasoning While Chain-of-Thought (CoT) prompting (Wei et al., 2022) has enhanced LLM reasoning, a significant performance gap persists in non-English contexts, particularly in mathematical tasks (Shi et al., 2022; Ahuja et al., 2023; Wang et al., 2025c). To mitigate this disparity, English-pivot strategies that translate inputs or reasoning processes into English have been proposed to leverage the models' strong English priors (Qin et al., 2023; Huang et al., 2023; Liu et al., 2025; Etxaniz et al., 2024). However, this approach often sacrifices interpretability for non-English users by generating final answers in English, thereby limiting practical utility in native-language applications (Wang et al., 2025b; Lee et al., 2025; Qi et al., 2025b).

Structured Reasoning Research on structure-aware prompting has evolved from Zero-shot CoT (Kojima et al., 2022) to more sophisticated approaches such as Skeleton-of-Thought (Ning et al., 2023), Plan-and-Solve (Wang et al., 2023), and Program-of-Thought (Chen et al., 2022). These methods have demonstrated that explicitly structuring reasoning enhances logical consistency (Zhou et al., 2022; Yao et al., 2023; Khot et al., 2022). While efforts have been made to extend structured reasoning to multilingual contexts through cross-lingual instruction tuning (Chai et al., 2025; Lai and Nissim, 2024), prior work has predominantly fixed the skeleton language to English, overlooking the impact of language combinations between the structural guide and the final response (Qi et al., 2025c; Li et al., 2024).

In this study, we refer to such structured guides as a *skeleton* and adopt the inference structure of Qi et al. (2025c) as a baseline to ensure experimental control. By fixing the structure and manipulating only the skeleton's language, we systematically

analyze how language configuration affects multilingual reasoning performance.

3 Language-Aware Skeleton Exploration Framework

This study investigates how the effectiveness of skeleton-based reasoning varies depending on language choice and evaluation contexts within a multilingual environment. To this end, we propose the *Language-Aware Skeleton Exploration Framework (*LASEF*)*, as illustrated in Fig. 1, which systematically analyzes skeleton utilization by varying language configurations while keeping the structure and format of the skeleton fixed. Here, "exploration" does not refer to new training or algorithmic search, but rather to an empirical analysis process designed to identify effective language configurations within a given evaluation environment.

3.1 Problem Setup and Language Dimensions

Standard language model inference typically involves a query q and a corresponding answer a, where the query is presented in a specific language ℓ_q. In skeleton-based reasoning, the model first generates a skeleton s, summarizing the core structure required for problem-solving based on the query, and subsequently uses it to generate the detailed reasoning process and the final answer. This process entails three distinct language choices given a user's target language ℓ_t: (1) the language of the query, ℓ_q; (2) the language of the skeleton, ℓ_s; and (3) the language in which the detailed reasoning and final answer are generated, ℓ_a.

These three languages may be identical or distinct; depending on their combination, the linguistic priors and reasoning paths elicited by the model can vary. Thus, a single skeleton-based reasoning process is characterized by the language combination (ℓ_q, ℓ_s, ℓ_a). LASEF focuses on isolating and analyzing the impact of these language configurations on reasoning performance while fixing the skeleton generation method and structure. Under this formulation, the skeleton-based reasoning process can be expressed as:

$$\hat{a} = f\big(q^{(\ell_q)}, s^{(\ell_s)}, \ell_a\big),$$

where $f(\cdot)$ represents the inference function of the pre-trained language model, and $s^{(\ell_s)}$ denotes the skeleton expressed in language ℓ_s.

Notably, the effective language combination for skeleton utilization is not universally fixed but may vary depending on the benchmark or domain used for evaluation. Given a specific benchmark B consisting of query-answer pairs $(\{q, a\} \in B)$, LASEF provides an analytical framework to identify relatively effective skeleton language configurations within that evaluation environment by comparing performance variations under different language combinations (ℓ_q, ℓ_s, ℓ_a).

This problem setup assumes a *training-free* environment involving no additional training or parameter updates. The following section describes the specific language configurations and skeleton usage methods considered in this study.

3.2 Skeleton Construction and Usage

The skeletons used in this study are intermediate representations that concisely reveal the *reasoning structure* for problem-solving, rather than presenting core answers or calculation procedures. In other words, the skeleton provides a minimal structural guide for the reasoning process to follow, avoiding specific instructions for the actual solution steps. This perspective conceptually aligns with existing research interpreting skeletons as structural representations that organize reasoning, rather than as summaries or hints for the solution.

Accordingly, our skeletons maintain a very short and abstract form, excluding specific calculation steps or answers. To ensure this, we consistently apply the following principle during skeleton generation: *"Provide only the essential structural characteristics of the problem. Describe what the problem structure is, never how to solve it."*

Fig. 1 presents a representative example of a skeleton generated in English ($\ell_s = en$) for a Swahili query ($\ell_q = sw$). Adhering to our proposed criteria, skeletons maintain a concise length of approximately 120 tokens on average regardless of the language, devoid of direct answers or core solution steps. We provide a detailed quantitative analysis of skeleton length and content distribution, along with the complete generation rules and additional examples, in Appendix A.

4 Experiments

In this section, we systematically validate the effectiveness of the proposed *Language-Aware Skeleton Exploration Framework (*LASEF*)* across diverse settings. The experiments are designed to analyze performance by varying model scale, problem difficulty, and language resource levels, while keeping

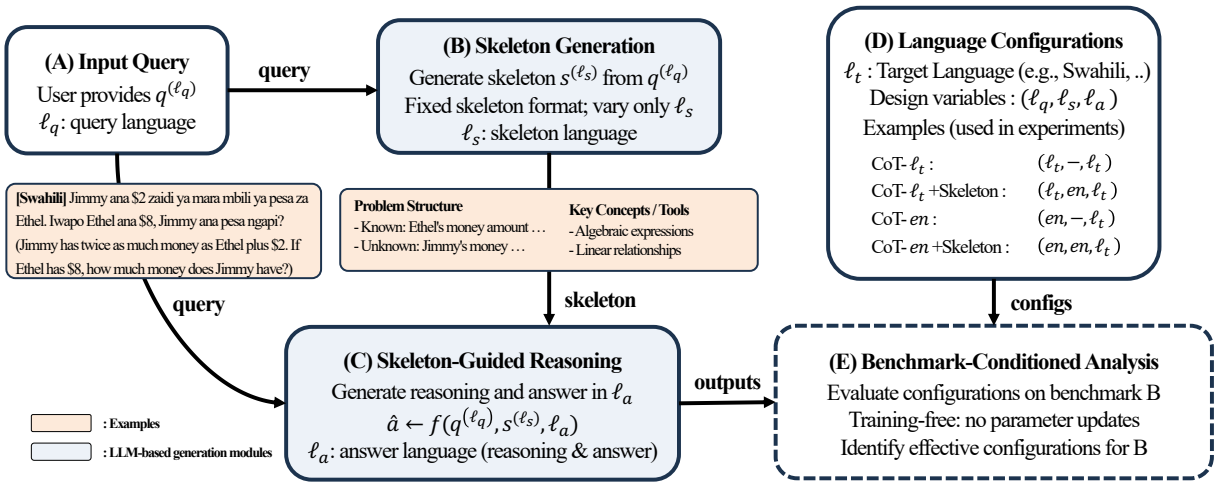


Figure 1: Overview of the Skeleton-Guided Reasoning Framework. (A) The process begins with an input query $q^{(\ell_q)}$. (B) A structured skeleton $s^{(\ell_s)}$ is generated to abstract the problem logic. (C) The final reasoning and answer $\hat{a}$ are produced in the answer language ℓ_a, conditioned on the query and skeleton. (D) We define the language configuration space as (ℓ_q, ℓ_s, ℓ_a) and compare four primary strategies: standard CoT and skeleton-guided approaches, applied in both the target language (ℓ_t) and English (en). (E) Effective configurations are identified through training-free benchmark analysis.

the structure and generation method of the skeleton fixed. Through this, we analyze (1) whether the effectiveness of skeleton-based reasoning is limited to specific conditions, and (2) how it varies according to language and model characteristics.

4.1 Experimental Setup

Models based on Scale. To analyze the impact of model scale, we select the QWEN2.5 series (Qwen et al., 2025) as our base models, following prior research (Qi et al., 2025c). These models are chosen for their superior multilingual capabilities as general-purpose models, rather than being specialized solely for reasoning. We evaluate 7B, 14B, and 72B models to examine how skeleton guidance scales. For brevity, we refer to the QWEN2.5-INSTRUCT models by their sizes (e.g., QWEN2.5-14B). Results for the LLAMA 3.1 series (Grattafiori et al., 2024) are provided in Appendix G.

Benchmarks by Difficulty. We categorize our benchmarks by difficulty to assess skeleton effectiveness across task complexities. We employ MGSM (Shi et al., 2022) for elementary arithmetic, MATH-500 (Lightman et al., 2023) for intermediate-level problems, and PolyMath (Wang et al., 2025c) for advanced reasoning tasks requiring complex structural analysis.

Target Languages based on Resource Levels. To evaluate the impact of skeletons according to language resource levels, we included diverse target languages, from high-resource to low-resource. High-resource languages were selected based on their sufficient exposure during large-scale pretraining, while low-resource languages were chosen based on the relative scarcity of training data, which often leads to inconsistent reasoning performance in multilingual LLMs. This allows us to analyze whether skeletons play different roles depending on the resource level of the language.

Evaluation Protocol. We use Exact Match (EM) as the evaluation metric, restricted to samples satisfying $\ell_a = \ell_t$ to ensure multilingual reasoning validity; reported scores are therefore *conditional reasoning accuracies* on language-compliant samples. Alongside greedy decoding, we conduct multi-rollout evaluation (T=0.7, N=5) to assess the stability of skeleton-language effects. Details on language identification, filtering and retention rates, and pairing are in Appendix B, and unfiltered results are in Appendix L.

4.2 Baseline Configurations

To analyze the effectiveness of LASEF, we define the language combination (ℓ_q, ℓ_s, ℓ_a) for a given target language ℓ_t, and establish an appropriate comparative baseline. Since skeletons function as structural guides for improving reasoning performance, we adopt a Chain-of-Thought (CoT)-based setting as the primary baseline (Wei et al., 2022), following prior work on multilingual reasoning (Huang et al., 2023; Qin et al., 2023).

Target-Language CoT (CoT-ℓ_t). In this setting, the input query is given in the target language, and both reasoning and the answer are generated in the

Method	Language	MGSM						MATH-500						PolyMath					
	(ℓ_q, ℓ_s, ℓ_a)	zh	es	ko	th	sw	AVG.	zh	es	ko	th	sw	AVG.	zh	es	ko	th	sw	AVG.
Qwen2.5-7B-Instruct																			
CoT-ℓ_t	$\ell_t, -, \ell_t$	79.9	80.7	69.7	80.2	16.0	65.3	56.4	57.5	47.8	49.8	14.2	45.1	30.8	31.0	26.4	26.1	5.1	23.9
+ SKELETON	ℓ_t, en, ℓ_t	80.7	83.8	71.0	79.0	22.9	67.5	55.4	57.7	49.3	51.9	20.4	46.9	30.6	32.1	28.0	27.6	7.4	25.1
CoT-$en^{\dagger}$	$en, -, \ell_t$	82.3	84.3	74.5	78.1	63.1	76.5	56.0	57.4	51.6	49.5	24.4	47.8	29.7	32.1	27.5	29.8	20.3	27.9
+ SKELETON	en, en, ℓ_t	82.3	85.9	73.2	78.9	71.7	78.4	57.4	58.2	52.3	53.8	29.8	50.3	32.0	34.4	27.7	30.4	19.9	28.9
Qwen2.5-14B-Instruct																			
CoT-ℓ_t	$\ell_t, -, \ell_t$	88.8	90.0	77.9	87.6	45.0	77.9	58.3	62.6	54.7	57.4	30.1	52.6	31.8	36.7	31.6	30.9	14.7	29.1
+ SKELETON	ℓ_t, en, ℓ_t	88.0	90.0	80.7	85.9	50.7	79.1	59.4	64.0	56.9	59.9	35.2	55.1	34.1	36.9	32.2	33.3	15.0	30.3
CoT-$en^{\dagger}$	$en, -, \ell_t$	83.1	88.3	81.2	84.8	74.5	82.4	58.8	60.2	53.8	55.1	39.1	53.4	32.3	36.5	30.1	32.0	24.0	31.0
+ SKELETON	en, en, ℓ_t	86.8	87.9	82.5	84.8	74.5	83.3	59.6	62.2	55.2	55.8	40.4	54.6	33.7	35.4	32.2	32.0	26.9	32.0
Qwen2.5-72B-Instruct																			
CoT-ℓ_t	$\ell_t, -, \ell_t$	88.3	91.6	83.9	90.8	68.0	84.5	63.1	63.8	63.2	61.6	46.3	59.6	36.1	37.5	34.5	34.7	31.0	34.8
+ SKELETON	ℓ_t, en, ℓ_t	88.0	89.2	84.7	89.6	70.1	84.3	65.8	63.8	62.4	63.4	54.3	61.9	37.1	37.5	37.0	35.0	32.0	35.7
CoT-$en^{\dagger}$	$en, -, \ell_t$	85.5	91.5	82.7	86.3	85.5	86.3	62.2	64.4	63.1	61.7	56.3	61.5	34.2	35.8	35.3	35.8	40.6	36.3
+ SKELETON	en, en, ℓ_t	86.8	90.7	83.1	86.7	86.7	86.8	64.2	64.0	61.1	62.5	60.0	62.4	34.0	37.9	36.2	33.1	39.3	36.1

Table 1: Results on MGSM, MATH-500, and PolyMath. (ℓ_q, ℓ_s, ℓ_a) denote the query, skeleton, and answer languages, respectively. CoT-$en^{\dagger}$ indicates queries translated into English using Google Translate, and the skeleton language is fixed to English unless otherwise noted. 'AVG.' denotes the average over five languages (zh, es, ko, th, sw). Greedy decoding throughout.

same language ($\ell_q = \ell_a = \ell_t$). We use a standard CoT prompt ("Let's think step by step, Respond in ℓ_t") without applying a skeleton. This represents the most natural usage scenario where users verify both the query and the reasoning process in their native language.

English-Pivot CoT (CoT-en). In this setting, the input query is translated into English ($\ell_q = \text{en}$), after which the reasoning and answer are generated in the target language ℓ_t. This baseline is essential for two reasons. First, comparing performance based on English-translated queries is a widely used practice in existing multilingual reasoning studies (Qin et al., 2023; Huang et al., 2023; Liu et al., 2025), representing the upper-bound performance of LLMs pre-trained primarily on English. Second, even in real-world non-English user environments, since users already understand the meaning of the query, internally converting the query to English does not hinder meaning transmission. Thus, CoT-en serves as a rational baseline to evaluate whether performance gains via translation are achievable without structural guidance.

Skeleton Baseline Language Configuration. English skeletons are used in Tab. 1 and 2, while Fig. 2 and 3 expands to Chinese, Spanish, Russian, Korean, and Thai.

4.3 Performance across Diverse Dimensions

Tab. 1 compares LASEF in an English skeleton setting (ℓ_q, en, ℓ_a) against baselines across three benchmarks: MGSM, MATH-500, and PolyMath.

Bridging the Resource Gap Language resource levels significantly influence the impact of English skeletons. Gains are substantially larger for low-resource languages like Swahili (sw) compared to high-resource languages like Chinese (zh) and Spanish (es). For instance, with QWEN2.5-7B on MGSM (CoT-$en^{\dagger}$), Swahili performance rose from 63.1 to 71.7 (+8.6), while Chinese and Spanish showed marginal gains (+0.0 and +1.6). This trend persists across scales; with QWEN2.5-72B on MATH-500, Swahili again saw the highest improvement (+3.7). These results indicate that when low-resource languages lack sufficient reasoning stability, English skeletons serve as an essential scaffold to stabilize the process.

Does Scaling Always Guarantee Improvement? With respect to model scale, we observe that smaller models benefit more from English skeletons, while the effect diminishes or even becomes negative as model size increases. QWEN2.5-7B and 14B exhibit performance improvements after applying skeletons across all benchmarks (+0.9 to 2.5). In contrast, QWEN2.5-72B shows an average performance drop of 0.2 after skeleton application under the CoT-ℓ_t setting on MGSM and the CoT-$en^{\dagger}$ setting on PolyMath. This suggests that for large-scale models, externally imposed English skeletons may act as an unnecessary constraint rather than a helpful inductive bias.

Does Benchmark Difficulty Limit the Gains? Examining performance variations across benchmark difficulty levels, we observe that the contribution of skeletons is larger for easier benchmarks. Using QWEN2.5-7B as the reference model, skeletons yield a performance gain of +2.2 on MGSM, the easiest benchmark, compared to +1.8 on MATH-500, which has medium difficulty, and +1.2

Language	ℓ_t, en, ℓ_t		$en^\dagger, en, \ell_t$	
	CoT-ℓ_t	+Skeleton	CoT-$en^\dagger$	+Skeleton
Qwen2.5-7B-Instruct				
kk	40.08	51.42 (+11.34)	62.82	72.22 (+9.40)
ky	33.33	37.86 (+4.53)	61.92	65.69 (+3.77)
mn	27.16	33.33 (+6.17)	66.96	64.78 (-2.18)
ug	27.13	37.65 (+10.52)	57.14	60.37 (+3.23)
hy	50.81	56.05 (+5.24)	64.49	73.88 (+9.39)
lo	25.51	23.48 (-2.03)	48.54	48.95 (+0.41)
eu	20.76	19.92 (-0.84)	65.62	74.55 (+8.93)
mt	25.65	20.00 (-5.65)	62.11	58.59 (-3.52)
am	8.23	10.29 (+2.06)	39.63	32.93 (-6.70)
ne	54.29	52.65 (-1.64)	74.58	78.81 (+4.23)
si	10.16	12.20 (+2.04)	37.14	35.92 (-1.22)
ps	26.46	26.91 (+0.45)	37.62	47.03 (+9.41)
sd	20.27	22.07 (+1.80)	64.81	68.52 (+3.71)
pa	51.41	52.21 (+0.80)	55.87	52.63 (-3.24)
gu	44.35	47.98 (+3.63)	59.84	60.64 (+0.80)
mr	45.34	46.96 (+1.62)	75.41	74.18 (-1.23)
or	44.35	43.95 (-0.40)	36.84	38.06 (+1.22)
ta	47.79	49.80 (+2.01)	69.48	69.08 (-0.40)
ml	42.74	46.37 (+3.63)	62.25	64.66 (+2.41)
AVG.	33.99	36.37 (+2.38)	58.06	60.08 (+2.02)

Table 2: Performance comparison on 19 low-resource languages (MGSM) using Qwen2.5-7B with English Skeleton. CoT-$en^\dagger$: English-translated queries via Google Translate. Parentheses show Δ gains.

on PolyMath, the most challenging benchmark. This trend suggests that as problem difficulty increases, simple skeleton generation alone becomes insufficient, as harder problems require more sophisticated and precise computational reasoning beyond what skeletons can provide.

Summary of Results. Results show that English skeletons' benefits depend on model scale, task difficulty, and language resources. Gains are most pronounced for smaller models, easier tasks, and low-resource languages. Notably, the CoT-$en^\dagger$ setting consistently outperforms CoT-ℓ_t, reaffirming the strong English prior of LLMs. This underscores translation as a critical baseline. In low-resource settings, most of the improvement comes from translation rather than from skeletons (+24.07 vs. +2.02 pp in Tab. 2).

4.4 Effect of English Skeletons in Low-Resource Languages

Previous analysis confirms that English skeletons are most effective for smaller models, easier benchmarks, and low-resource languages. In this section, we validate this finding by evaluating Qwen2.5-7B on 19 low-resource MGSM languages. This section examines the consistency of skeleton-based CoT improvements under these conditions. The criteria for language selection and our translation methodology are detailed in Appendix C.

Tab. 2 summarizes the performance changes under the target-language input (ℓ_t) and English-translated input ($en^\dagger$) settings. English skeletons show a small positive tendency on average: applying them improves 13 of the 19 languages, with mean gains of +2.38 in the CoT-ℓ_t setting and +2.02 in the CoT-$en^\dagger$ setting. However, after FDR correction, statistically significant language-level improvements are observed only for Kazakh (kk) and Uyghur (ug); pooling the 19 languages at the item level, the overall effect remains significant (Appendix J).

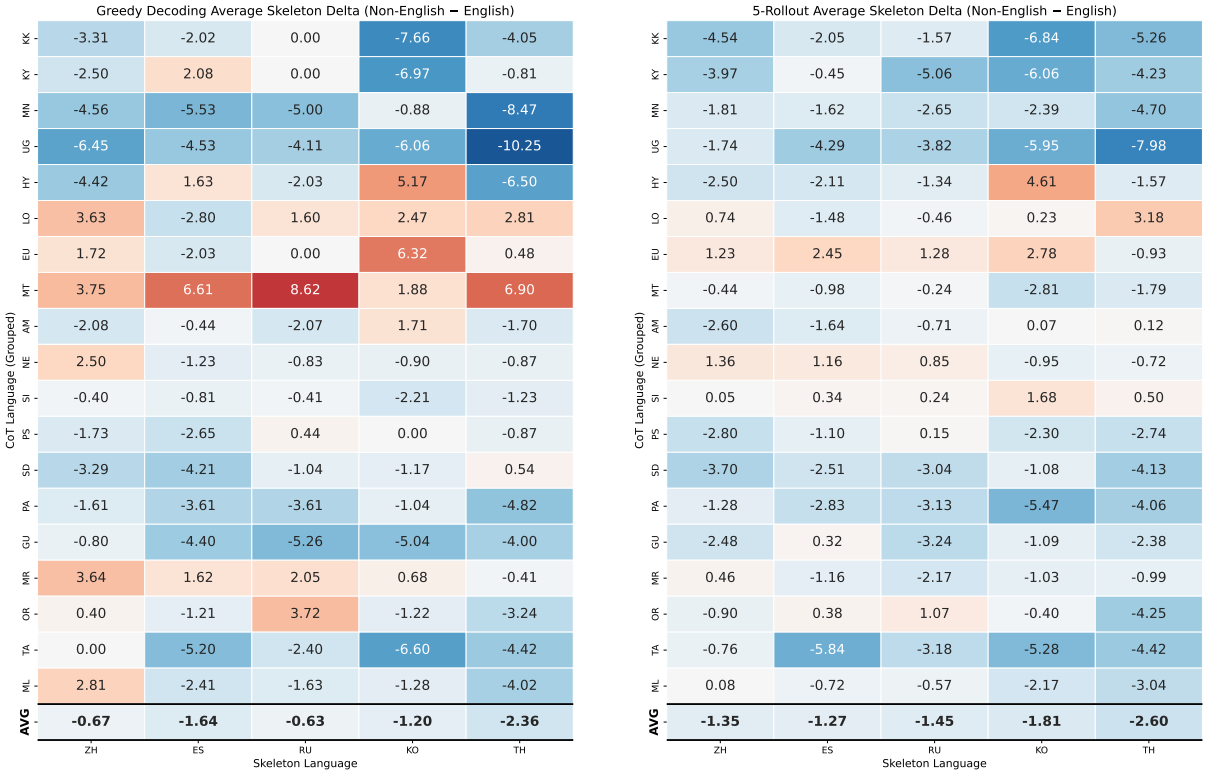
First, under the Native Input (CoT-ℓ_t) setting, skeletons substantially unlock the model's latent reasoning capacity for certain languages, such as Kazakh (kk) (+11.34) and Uyghur (ug) (+10.52). However, there also exist cases where performance degrades, such as Maltese (mt) (5.65), indicating a high variance in language-specific effects.

Second, under the (CoT-$en^\dagger$) setting, the effect of English skeletons becomes considerably more consistent in direction. Notably, query translation alone leads to a dramatic increase in average performance, from 33.99 to 58.06. When English skeletons are incorporated, an additional performance gain of +2.02 is observed. In particular, languages such as Armenian (hy) (+9.39), Basque (eu) (+8.93), and Pashto (ps) (+9.41) exhibit substantial improvements when structure in the form of skeletons is added on top of translated queries.

Overall, in low-resource language settings, query translation provides the dominant improvement, and English skeletons contribute a smaller additional gain by structuring the reasoning process. This suggests that skeleton-based CoT can serve as a useful supplementary reasoning aid—rather than the main driver of improvement—even in extremely low-resource language scenarios.

4.5 Is English the Only Solution for Low-Resource Languages?

In Sec. 4.4, we showed that English skeletons generally serve as effective reasoning aids. However, the CoT-ℓ_t results in Tab. 2 indicate that English skeletons are not always beneficial for all low-resource languages. In particular, five out of the 19 languages (lo, eu, mt, ne, and or) show performance degradation when English skeletons are applied (the 19 languages are those of the 34 MGSM low-resource languages that pass the sample filter; see Appendix B.2).



(a) Greedy Decoding (single rollout)

(b) Multi-rollout (T=0.7, N=5)

Figure 2: Performance differences (Non-English − English) on MGSM under CoT-ℓ_t with QWEN2.5-7B. The English skeleton is replaced with five Non-English skeletons (ZH, ES, RU, KO, TH); the same skeleton set is used in both panels. **(a)** Greedy decoding (single rollout). **(b)** Multi-rollout sampling (T=0.7, N=5). The corresponding English-skeleton reference values (English skeleton vs. no skeleton per language) are reported in Tab. 2.

In this section, we extend LASEF to five non-English skeletons (zh, es, ru, ko, and th) and analyze how the effect of skeleton language choice varies across target languages. We distinguish surface *affinity* (language family, script, and geographic proximity) from structural *alignment* (typology and word order), and examine the stability of the observed effects using both greedy decoding and multi-rollout sampling.

General Trends: English as a Stable Anchor. Fig. 2 shows the performance differences between non-English and English skeletons under the CoT-ℓ_t setting with QWEN2.5-7B, using greedy decoding and multi-rollout evaluation. Overall, applying non-English skeletons uniformly leads to an average performance decrease across languages, ranging from 0.63 to 2.60 points. This result reaffirms that English serves as a stable reasoning skeleton for most languages. However, when examining individual languages, notable exceptions emerge, particularly for languages on which the English skeleton had a negative effect.

Directionally Consistent Cross-Lingual Effects. As shown in Tab. 2, some languages that experience performance degradation with English skeletons exhibit clear improvements when paired with alternative skeletons that share affinity or alignment with the target language. Lao (lo) suffers a performance drop with the English skeleton (-2.03), but shows a $+2.81$ improvement with a Thai skeleton, which shares surface affinity through the Tai–Kadai language family (Fig. 2(a)). Similarly, Basque (eu) and Armenian (hy) achieve gains of $+6.32$ and $+5.17$, respectively, when paired with a Korean skeleton, which shares their structural alignment (agglutinative morphology and SOV word order). Crucially, *the sign of these effects is preserved under multi-rollout evaluation* (Fig. 2(b)): hy–ko slightly decreases ($+5.17 \rightarrow +4.61$), lo–th is preserved ($+2.81 \rightarrow +3.18$), and eu–ko also maintains a positive effect ($+6.32 \rightarrow +2.78$). This consistency suggests that these effects are not artifacts of a specific decoding setting, but directionally consistent patterns that recur across certain language combinations (statistical significance and cross-benchmark checks are reported in Appendices J and K).

Case Study: Maltese. Another notable case is Maltese (mt). In Fig. 2(a), Maltese shows large positive effects with Russian (+8.62), Thai (+6.90), and Spanish (+6.61) skeletons, and all three pairs satisfy nominal significance under McNemar's test (Appendix J). However, these effects are attenuated or even reversed under multi-rollout evaluation (e.g., mt–ru: −0.23). Therefore, unlike the directionally consistent cases discussed above, the positive effects observed for Maltese are sensitive to the evaluation protocol and to the benchmark (Appendix K), and we do not treat them as evidence of a generalizable positive effect. Since the results in this section alone are insufficient to determine the source of this pattern, we further examine it in Sec. 5.

Asymmetric Negative Effects Surface-level similarities such as shared scripts or geographic proximity do not necessarily lead to positive transfer. Mongolian (mn) and Russian share the Cyrillic script, but differ in word order and morphological structure: Mongolian is an agglutinative language, whereas Russian is fusional. Accordingly, Russian skeletons lead to performance degradation under both greedy decoding (−5.00) and multi-rollout evaluation (−2.65). Similarly, Uyghur (ug) and Chinese are geographically proximate but structurally different: Uyghur is an SOV agglutinative language, whereas Chinese is an SVO analytic language. Chinese skeletons degrade Uyghur performance (−6.45 greedy; −1.12 multi-rollout). These cases suggest that surface-level similarity alone is insufficient to predict the direction of skeleton transfer; this negative pattern largely persists on MSVAMP (Appendix K).

The Necessity of LASEF. Taken together, the effects of skeleton language fall into three patterns: (1) *directionally consistent effects*, where the sign of positive transfer holds under both greedy and multi-rollout evaluation; (2) *evaluation- and benchmark-dependent effects*, where large greedy gains are attenuated under multi-rollout evaluation or on an unseen benchmark; and (3) *asymmetric negative effects*, where negative transfer occurs despite shared surface affinity. These heterogeneous, largely exploratory patterns (Appendices J and K) show that a single evaluation protocol is insufficient to fully characterize skeleton-language effects, supporting the need for LASEF as a multi-level empirical analysis framework rather than a rule-based selector.

5 Analysis

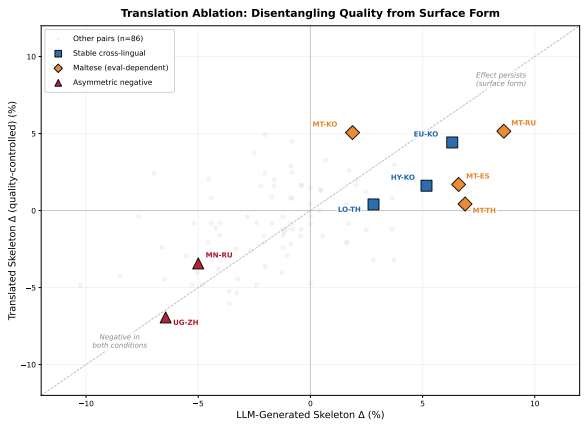


Figure 3: Translation ablation results. Each point is a (target, skeleton) pair, with Δ from LLM-generated skeletons on the x-axis and translated skeletons on the y-axis. Points near the diagonal indicate effects robust to skeleton quality.

The effects observed in Sec. 4.5 may conflate two factors: differences in skeleton *generation quality* and *cross-lingual alignment* between skeleton and target language. To disentangle these factors, we conduct a translation ablation. Specifically, we translate the English skeleton into five non-English languages (zh, es, ru, ko, th) using GPT-5-mini, and then run greedy-decoding inference with the translated skeletons, comparing them against the English-skeleton baseline.

Since the semantic content is fixed, any observed difference more directly reflects surface-form variation rather than skeleton generation quality. Fig. 3 visualizes the results: the x-axis denotes Δ with LLM-generated skeletons, the y-axis denotes Δ with translated skeletons, and points near the diagonal (y=x) indicate effects that are less sensitive to skeleton generation quality. The qualitative LLM-as-a-Judge analysis is provided in Appendix F.

Directionally Consistent Pairs Persist The directionally consistent pairs from Sec. 4.5 fall in the upper-right region of Fig. 3 and retain a positive sign under translation ablation: Basque–Korean (+6.32 → +4.42), Armenian–Korean (+5.17 → +1.61), and Lao–Thai (+2.81 → +0.40). Although the magnitude of Δ decreases, the positive sign is preserved, suggesting that these effects are not solely attributable to skeleton generation quality.

Maltese: A Fragile Surface Effect Maltese is the clearest evaluation-dependent case: large greedy gains with several non-English skeletons

are attenuated or reversed under multi-rollout evaluation. The translation ablation shows this is not a generation-quality artifact: with content fixed and only the surface language varied, the gains stay positive (mt–ru: $+5.15$, mt–ko: $+5.06$). It is thus a genuine surface-form effect, but one sensitive to the decoding protocol and to the evaluation benchmark: on MSVAMP, the mt–ru gain reverses (Appendix K).

Asymmetric Negatives Persist In contrast, the asymmetric negative pairs, Mongolian–Russian (-3.43) and Uyghur–Chinese (-6.94), lie near the diagonal in the lower-left quadrant of Fig. 3: both LLM-generated and translated skeletons yield comparable negative transfer, so it is not a quality artifact. These pairs share surface affinity (script or geography) but lack structural alignment (Appendix D), so the effect tracks misalignment, not quality or surface similarity.

Together with the stable and Maltese pairs, this shows that skeleton-language effects are not reducible to generation quality but reflect genuine skeleton–target interactions—effects that cannot be predicted a priori and must be found empirically, the role of LASEF.

6 Conclusion

We analyzed the problem of skeleton language selection in multilingual mathematical reasoning through the Language-Aware Skeleton Exploration Framework (LASEF), an empirical analysis framework rather than an automatic selector of an optimal skeleton language. Our experiments show that English skeletons are a reasonable default, but their average gains are small and they are not a one-size-fits-all solution. By combining greedy decoding, multi-rollout evaluation, translation ablation, and cross-benchmark validation, we further find that skeleton-language effects can be categorized into three patterns: directionally consistent, evaluation- and benchmark-dependent, and asymmetric negative. These results answer our research question conditionally rather than universally: skeleton language is a context-dependent design variable that cannot be fully explained by generation quality alone, and effective—or risky—language combinations should be identified through multi-level exploration.

Limitations

Our study has several limitations. First, we focused exclusively on mathematical reasoning; generalizing findings to domains like commonsense reasoning or code generation requires further validation. Second, our training-free approach ensures broad applicability but precludes potential gains from fine-tuning. Third, we maintained a fixed skeleton structure to isolate language effects, leaving the interplay between skeleton design and language choice unexplored. Finally, while we covered diverse languages, the full LASEFconfiguration search remains computationally expensive and is intended as an offline diagnostic rather than a per-deployment step; results from an existing benchmark can help flag risky skeleton candidates but should be revalidated whenever the benchmark, model, or task changes (Appendix K). Developing efficient predictive methods for this search remains future work.

Acknowledgments

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (No. 2022-0-00871, Development of AI Autonomy and Knowledge Enhancement for AI Agent Collaboration; No. RS-2026-25525363, AEGIS: Agentic Experts for Generative-AI Inspection Solution). This work also utilized GPU resources from the "Advanced GPU Utilization Support Program" funded by MSIT, Republic of Korea (awarded to KyungTae Lim).

References

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, and 1 others. 2023. Mega: Multilingual evaluation of generative ai. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267.

Linzheng Chai, Jian Yang, Tao Sun, Hongcheng Guo, Jiaheng Liu, Bing Wang, Xinnian Liang, Jiaqi Bai, Tongliang Li, Qiyao Peng, and 1 others. 2025. xcot: Cross-lingual instruction tuning for cross-lingual chain-of-thought reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23550–23558.

Nuo Chen, Zinan Zheng, Ning Wu, Ming Gong, Dongmei Zhang, and Jia Li. 2024. Breaking language

barriers in multilingual mathematical reasoning: Insights and observations. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7001–7016.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Julen Etxaniz, Gorka Azkune, Aitor Soroa, Oier Lopez de Lacalle, and Mikel Artetxe. 2024. Do multilingual language models think better in english? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 550–564.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Haoyang Huang, Tianyi Tang, Dongdong Zhang, Wayne Xin Zhao, Ting Song, Yan Xia, and Furu Wei. 2023. Not all languages are created equal in llms: Improving multilingual capability by cross-lingual-thought prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12365–12394.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed prompting: A modular approach for solving complex tasks. In *The Eleventh International Conference on Learning Representations*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Hynek Kydlíček. Math-Verify: Math Verification Library.

Huiyuan Lai and Malvina Nissim. 2024. mcot: Multilingual instruction tuning for reasoning consistency in language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12012–12026.

Nahyun Lee, Yeongseo Woo, Hyunwoo Ko, and Guijin Son. 2025. Controlling language confusion in multilingual llms. In *ACL 2025 Student Research Workshop*.

Bryan Li, Tamer Alkhouli, Daniele Bonadiman, Nikolaos Pappas, and Saab Mansour. 2024. Eliciting better multilingual structured reasoning from llms through code. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5154–5169.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, and 101 others. 2026. Ministral 3. *Preprint*, arXiv:2601.08584.

Chaoqun Liu, Wenxuan Zhang, Yiran Zhao, Luu Anh Tuan, and Lidong Bing. 2025. Is translation all you need? a study on solving multilingual tasks with large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 9594–9614.

Xuefei Ning, Zinan Lin, Zixuan Zhou, Zifu Wang, Huazhong Yang, and Yu Wang. 2023. Skeleton-of-thought: Large language models can do parallel decoding. *Proceedings ENLSP-III*.

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, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Jirui Qi, Shan Chen, Zidi Xiong, Raquel Fernández, Danielle Bitterman, and Arianna Bisazza. 2025a. When models reason in your language: Controlling thinking language comes at the cost of accuracy. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 20279–20296.

Jirui Qi, Shan Chen, Zidi Xiong, Raquel Fernández, Danielle S Bitterman, and Arianna Bisazza. 2025b. When models reason in your language: Controlling thinking language comes at the cost of accuracy. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 20279–20296.

Rui Qi, Zhibo Man, Yufeng Chen, Fengran Mo, Jinan Xu, and Kaiyu Huang. 2025c. Sot: Structured-of-thought prompting guides multilingual reasoning in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 11024–11039.

Shenbin Qian, Archchana Sindhujan, Minnie Kabra, Diptesh Kanojia, Constantin Orasan, Tharindu Ranasinghe, and Fred Blain. 2024. What do large language models need for machine translation evaluation? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3660–3674.

Libo Qin, Qiguang Chen, Fuxuan Wei, Shijue Huang, and Wanxiang Che. 2023. Cross-lingual prompting: Improving zero-shot chain-of-thought reasoning across languages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2695–2709.

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

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, and 1 others. 2022. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, and 1332 others. 2025. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Bin Wang, Zhengyuan Liu, Xin Huang, Fangkai Jiao, Yang Ding, AiTi Aw, and Nancy Chen. 2024. 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)*, pages 370–390.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634.

Mingyang Wang, Lukas Lange, Heike Adel, Yunpu Ma, Jannik Strötgen, and Hinrich Schuetze. 2025a. Language mixing in reasoning language models: Patterns, impact, and internal causes. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2637–2665, Suzhou, China. Association for Computational Linguistics.

Qihan Wang, Shidong Pan, Tal Linzen, and Emily Black. 2025b. Multilingual prompting for improving LLM generation diversity. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6378–6400, Suzhou, China. Association for Computational Linguistics.

Yiming Wang, Pei Zhang, Jialong Tang, Haoran Wei, Baosong Yang, Rui Wang, Chenshu Sun, Feitong Sun, Jiran Zhang, Junxuan Wu, Qiqian Cang, Yichang Zhang, Fei Huang, Junyang Lin, Fei Huang, and Jingren Zhou. 2025c. Polymath: Evaluating mathematical reasoning in multilingual contexts. *The Thirty-Ninth Annual Conference on Neural Information Processing Systems*.

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

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and 1 others. 2022. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.

Wenhao Zhu, Shujian Huang, Fei Yuan, Shuaijie She, Jiajun Chen, and Alexandra Birch. 2024. Question translation training for better multilingual reasoning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 8411–8423.

A Skeleton

A.1 Skeleton Generation Method

The skeleton generation pipeline proposed in this study is constructed as follows.

Input Configuration When generating a skeleton, the model is provided with the problem text together with three few-shot examples. These few-shot examples constrain the output style and abstraction level of the skeleton, ensuring that the skeleton is produced in the specified output language ℓ_s. Meanwhile, the input question itself is presented in the target language ℓ_t, and the model generates the skeleton based on this input.

Two Components Instead of directly presenting the computational steps or explicit reasoning procedures required to solve the problem, the skeleton is designed to abstract and represent the underlying structural elements and the conceptual problem-solving framework. The generated skeleton is organized into the following two core components:

1. **Problem Structure**: Identifies and defines the **Objects**, **Variables**, **Relationships**, **Constraints**, and **Problem Nature** included in the problem.

2. **Key Concepts / Tools**: Describes the mathematical/scientific principles and tools essential for solving the problem using standardized academic terminology.

Generation Constraints To maintain the level of abstraction and consistency of the skeleton, the generation process is subject to the following constraints. First, the final answer or any specific numerical values are not mentioned explicitly, and the numbers or formulas appearing in the reasoning path are replaced with their conceptual roles (e.g., "coefficient", "boundary value", "ratio"). Second, instead of computation-centered verbs such as "calculate" or "compute" the model is encouraged to use verbs that induce conceptual reasoning, such as "identify", "express", and "relate". Third, while the essential logical structure of the problem-solving process is preserved, peripheral procedural details, redundant descriptions, and errors contained in the reasoning path are excluded. Through these constraints, the skeleton is guided to extract the structural essence of the problem without directly presenting the solution method.

A.2 System Prompt for Skeleton Generation

The details of the System Prompt applied to the model for skeleton generation are presented in Tab. 3. This prompt is designed to ensure that the model does not directly present the final answer or explicit solution procedures (How), but instead abstracts the structural characteristics and core concepts (What) of the problem to construct a conceptual problem-solving plan. The prompt used in the subsequent stage, where the model solves the problem based on the generated skeleton, is presented as follows.

Solver Prompt for Skeleton

Question: [Problem Text]
Reasoning Skeleton: [Generated Skeleton]
Instructions:
1. Follow the Reasoning Skeleton above.
2. Verify numbers against the Question text.
3. Let's think step-by-step. Respond in [Target Language].
Final Answer Format:

answer

A.3 Few-shot Examples Used for Skeleton Reasoning

Skeleton Few-shot Example 1 (Korean)

Question: 삼각형에 길이가 5cm와 12cm인 두 변이 있으며, 그 사이에 직각이 있습니다. 세 번째 변의 길이는 얼마입니까? (A triangle has sides of length 5 cm and 12 cm, with a right angle between them. What is the length of the third side?)

Skeleton:
Problem Structure
- Geometric figure: right triangle with three sides
- Known quantities: two perpendicular sides (5 cm, 12 cm), right angle constraint
- Unknown: third side (hypotenuse) connecting the two legs
- Relationship: side lengths satisfy distance relation enforced by right angle property

Key Concepts / Tools
- Pythagorean theorem
- Right triangle geometry

System prompt for skeleton generation

Your Role:

You generate minimal, high-level reasoning skeletons for math and science problems.

Language Requirement:

Always respond in English, regardless of the language used in the user query.

Goal:

Provide only the *essential structural characteristics* of the problem. You must describe WHAT the problem structure is, never HOW to solve it.

Output Format: Provide exactly two sections below.

1. **Problem Structure**

Identify and characterize:

- **Objects & Variables**: Types of entities (quantities, sets, functions, geometric figures), separating constants/parameters from variables. Distinguish knowns vs. unknowns.
- **Relationships**: The nature of connections between entities (e.g., proportional, inverse, functional dependency, spatial). Describe the *type* of relationship, not the derivation.
- **Constraints**: Conditions defining the problem space (equations, inequalities, domain restrictions, boundary conditions, conservation laws).
- **Problem Nature**: (Only if relevant) Briefly note the structural type (e.g., Optimization, Combinatorial, Inverse problem).

Constraints for this section:

- Must be problem-specific, not generic.
- Must NOT include solution steps, strategies, formulas, or numerical manipulations.

2. **Key Concepts / Tools**

List relevant mathematical or scientific principles using standard terminology (e.g., “Pythagorean Theorem”, “Conservation of Energy”, “Bayes’ Theorem”).

Constraints for this section:

- Use established concept names only.
- Do NOT explain how to apply them or list formulas.

Table 3: The prompt instructs the model to produce a concise, high-level outline of the reasoning process, while omitting detailed computations and final answers.

Skeleton Few-shot Example 2 (Korean)

Question: 5명의 후보(Alice, Bob, Charlie, Dave, Eve) 중에서 2명의 위원을 선출해야 합니다. 그러나 Alice와 Bob은 함께 위원이 될 수 없습니다. 위원회를 구성하는 방법은 몇 가지입니까? (A committee needs to pick 2 members from a group of 5 candidates. However, Alice and Bob cannot be on the committee together. How many ways can the committee be formed?)

Skeleton:
Problem Structure
- Discrete selection problem: choosing k=2 elements from n=5 candidates
- Known: total candidates (5), committee size (2), pairwise exclusion constraint
- Unknown: number of valid selection configurations
- Constraint: pairwise exclusion (Alice and Bob cannot co-occur)

Key Concepts / Tools
- Combinatorial counting (combinations)
- Constraint satisfaction
- Complementary counting

Few-shot Example 3: Rate Problem

Question: 기계 A는 8시간 동안 시간당 100개를 생산합니다. 기계 B는 4시간 가동 후 1시간 휴식하고, 그 후 3시간 동안 시간당 120개를 생산합니다. 총 생산량은 얼마입니까? (Machine A runs for 8 hours producing 100 units/hour. Machine B runs for 4 hours, rests for 1 hour, then runs for 3 hours at 120 units/hour. What is the total production?)

Skeleton:
Problem Structure
- Two independent production processes with different temporal patterns
- Variables: production rates, time durations, total output
- Machine A: single continuous active interval (8 hours @ 100 units/hour)
- Machine B: two disjoint active intervals separated by idle period
- Relationship: total output = sum of (rate $\times$ active duration) across all segments

Key Concepts / Tools
- Rate-time-quantity relationships
- Piecewise continuous processes
- Linear aggregation

For each target language (Korean, Spanish, Chinese, Thai, Swahili), we used few-shot examples where the questions were translated into the respective language, while the skeleton responses were generated in English in all cases.

A.4 Example of Skeleton Generation

Example (1): Skeleton Generation

Question: 한 자전거가 분당 20피트의 속도로 이동하고 있다. 이 자전거의 속도를 초당 인치로 표현하면 얼마인가? (A bicycle is traveling at 20 feet per minute. What is the bicycle's speed expressed in inches per second?)

Skeleton:
Problem Structure
- Object: bicycle
- Known: speed in feet per minute (20 ft/min)
- Unknown: speed in inches per second
- Relationships: conversion between units of length (feet to inches) and time (minutes to seconds)
- Constraints: 1 foot = 12 inches, 1 minute = 60 seconds

Key Concepts / Tools
- Unit conversion
- Dimensional analysis

Example (2): Skeleton Generation

Question: 6명의 마녀와 10명의 마법사가 비밀 모임에 참석하였다. 마녀들은 각 마법사와 악수를 하기로 결정했다. 마녀들끼리는 이미 모두 친한 친구이므로 서로 악수를 하지 않고, 마법사들끼리는 서로 싫어해서 악수를 하지 않았다. 이 모임에서 이루어진 총 악수의 수는 몇인가? (Six witches and ten sorcerers are at an arcane mixer. The witches have decided to shake hands with each of the sorcerers. None of the witches bothered to shake each other's hands, since they are all good friends already, and the sorcerers all sort of hate each other and did not shake hands with other sorcerers. How many handshakes took place at the mixer?)

Skeleton:
Problem Structure
- Discrete interaction problem: handshakes between two distinct groups
- Known: number of witches (6), number of wizards (10)
- Unknown: total number of handshakes
- Relationship: each witch shakes hands with each wizard exactly once
- Constraints: no handshakes between witches, no handshakes between wizards

Key Concepts / Tools
- Combinatorial counting (specifically, Cartesian product of two sets)
- Bipartite graph theory (each handshake represents an edge between two disjoint sets)

B Evaluation Setup

B.1 Chain-of-Thought Prompt

We adopt Chain-of-Thought (Wei et al., 2022) as our baseline approach. The prompt is shown below.

Prompt for CoT

System:
You are a math problem solver. Solve the given problem.
At the end, clearly state the final answer in the following format:

final answer

User:
Question: [Problem Text]
Let's think step by step. Respond in [Target Language].

B.2 Evaluation Protocol

Decoding Strategy To minimize variability in the reasoning process across all experiments, Greedy Decoding was used. The temperature was set to 0, selecting the token with the highest probability at each generation step.

Exact Match (EM) Evaluation Quantitative evaluation was performed using the `math-verify` library (Kydlíček). The Exact Match (EM) score was calculated based on whether the answer extracted in the '`\boxed{}`' format from the model's final output matched the Ground Truth.

Language Consistency Filtering To ensure a fair comparison of reasoning capabilities, we included in the evaluation only those samples whose generated output language matched the target language. In particular, we filtered samples satisfying the condition $\ell_a = \ell_t$, thereby preventing evaluation bias that may arise from code-switching or unintended language mixing. Language identification for the generated responses was performed using the FastText classifier (Joulin et al., 2017). Applying this filter to the high-resource setting of Tab. 1 retains 98.8/98.4% (MGSM), 93.1/93.4% (MATH-500), and 92.5/92.4% (PolyMath) of samples for target-language/English-translated queries; retention varies more substantially across low-resource languages, and per-language sample counts together with unfiltered accuracies are reported in Appendix L.

In addition, to ensure a fair comparison between the baseline and the skeleton-based method (e.g., CoT-ℓ_t vs. +SKELETON), evaluation was conducted only on samples that satisfied the condition $\ell_q = \ell_t$ in both settings. This filtering ensured that the reasoning performance was compared under identical input–output language conditions.

Intersection-based Pairing. For comparisons across skeleton languages in Sec. 4.5, we adopt intersection-based pairing to ensure fair comparison. Specifically, each (target language, skeleton language) pair is compared against its English-skeleton counterpart only on the subset of items where both settings produce $\ell_a = \ell_t$. To ensure reliable analysis, we further require each target language to retain at least 180 valid samples on average across the five non-English skeleton languages (zh, es, ru, ko, th); target languages with fewer samples are excluded. In total, MGSM-LowResource covers 34 target languages; the 19 that satisfy this criterion form the main analysis set of Sec. 4.5, and combined with the five non-English skeleton languages they yield the 95 target–skeleton pairs tested in Appendix J.

Language Forcing Following the approach proposed by Qi et al. (2025a), we incorporate the instruction "Respond in $\{l_t\}$" along with language-specific trigger phrases. Specifically, we use triggers such as "좋습니다," for Korean, "Sawa," for Swahili, and "Bien," for Spanish to encourage the model to initiate generation in the target language.

B.3 Mathematical Benchmarks

In this experiment, to comprehensively evaluate the mathematical reasoning capabilities of multilingual LLMs, we utilized three benchmarks with differing problem difficulties and linguistic characteristics.

MGSM (Multilingual Grade School Math) A multilingual benchmark built upon the 250-sample test set of GSM8K (Cobbe et al., 2021) (Shi et al., 2022), consisting of elementary-level arithmetic problems. It is widely used as a standard metric for measuring basic arithmetic reasoning skills in multilingual environments. In addition to the existing 10 languages (Bengali, Chinese, English, French, German, Japanese, Russian, Spanish, Swahili, Thai), we expanded the evaluation scope to include Korean[2] and various low-resource languages.

MATH-500 A subset of 500 representative questions selected by topic and difficulty from the high-difficulty mathematics dataset MATH (Hendrycks et al., 2021). It covers advanced mathematical fields such as Algebra, Geometry, and Calculus, requiring complex mathematical thinking and problem-

[2] https://huggingface.co/datasets/freewheelin/mgsm_ko

solving skills beyond simple calculations. Since MATH is originally an English-only benchmark, we utilized existing high-quality human translations[3] for our multilingual evaluation.

PolyMath A recently proposed high-difficulty multilingual math reasoning benchmark (Wang et al., 2025c), containing 500 complex problems (125 per difficulty level) across four difficulty levels in 18 languages. Composed of high-quality questions reflecting the cultural and linguistic contexts of each language region rather than simple translations, it is suitable for analyzing the deep multilingual reasoning limitations and performance gaps of LLMs.

B.4 Models

We selected QWEN2.5 and LLAMA 3.1, the latest open-source model families with verified multilingual processing capabilities and reasoning performance.

Qwen2.5 (Qwen et al., 2025) Developed by Alibaba Cloud, the QWEN2.5 series has demonstrated exceptional performance, particularly in mathematical reasoning and coding. It offers various parameter sizes ranging from 0.5B to 72B and is pretrained on a vast multilingual corpus, exhibiting strong performance even in non-English languages. In this study, we used the 7B, 14B, and 72B models to analyze the effect of skeletons according to model scale expansion.

Llama 3.1 (Grattafiori et al., 2024) As the successor to Meta's LLAMA 3, LLAMA 3.1 features an extended context window of 128K tokens and enhanced multilingual support. To verify whether the skeleton-based methodology is effective in general-purpose LLMs not specialized for specific domains, we used the 8B and 70B models in our experiments.

C Details of MGSM Low-Resource Dataset Construction

This section details the construction process and rigorous quality control system of the **MGSM Low-Resource** benchmark newly established in this study. To expand the existing MGSM dataset to 34 low-resource languages, we introduced a Dual-Model Translation Pipeline applying prompt engineering and a multi-stage verification procedure.

C.1 Selection Criteria for Low-Resource Languages

Target languages were selected from ultra-low-resource languages that account for less than 0.05% of the data within the large-scale corpus `allenai/c4`[4]. This was to identify languages that are not sufficiently included in the pre-training data of modern LLMs, leading to underestimated performance or lack of research.

The 34 finally selected languages were chosen considering typological diversity, including various language families, writing systems, and geographical distributions. The detailed list is provided in Tab. 4.

Region	Languages	Script
Africa	Amharic	Ge'ez
Europe	Basque, Maltese, Armenian	Latin, Armenian
Central Asia	Kazakh, Kyrgyz, Uyghur, Pashto	Cyrillic, Arabic
South Asia	Nepali, Sinhala, Sindhi, Punjabi, Gujarati, Marathi, Odia, Tamil, Malayalam	Indic, Arabic
SE Asia & Pacific	Lao, Mongolian	Lao, Cyrillic

Table 4: List of the 19 low-resource target languages included in the MGSM-LowResource dataset.

C.2 Dataset Construction Pipeline

The dataset construction adopted a dual-model pipeline utilizing Gemini (`gemini-3.0-flash`) (Team et al., 2025) and GPT (`gpt-5-mini`) (OpenAI et al., 2024) in parallel to eliminate single-model dependency and maximize translation quality. The entire process consists of three stages: **Generation-Validation-Selection**.

C.2.1 Stage 1: Prompt-Based Generation and Numerical Consistency Check

For each problem, two models independently generate translation candidates (T). To prevent distortion of mathematical symbols or unnecessary additions and ensure translation accuracy, we designed and applied a system prompt including a professional translator persona and four key constraints (numerical preservation, meaning retention, natural fluency, and output format compliance).

The details of the prompt used are as follows:

[3] https://huggingface.co/datasets/appier-ai-research/Multilingual-MATH-500, https://huggingface.co/datasets/davidkim205/ko-math-500, https://huggingface.co/datasets/iapp/math-500-th

[4] https://huggingface.co/datasets/allenai/c4

Translation System Prompt

"You are a professional translator. Translate the given text to [Target Language].
Rules:
1. Preserve all numbers, mathematical expressions, and proper nouns exactly as they appear
2. Maintain the original meaning and context
3. Use natural, fluent [Target Language]
4. Only output the translation, nothing else"

Particularly, to guarantee numerical accuracy, which is core information in math word problems, a **Numerical Preservation** check was performed immediately after generation. We calculated the recall of numerical entities between the source text (E) and the translated text (T), and if it fell below the threshold (80%), the translation was immediately discarded, and the regeneration process was triggered.

$$\text{Recall}_{\text{num}}(T, E) < 0.8 \implies \text{Regenerate} \quad (1)$$

Through this strict filtering process, numerical errors were preemptively blocked, and only valid candidates adhering to prompt constraints were passed to the next stage.

C.2.2 Stage 2: Back-Translation Quality Selection (BTQS)

To select the optimal translation among the candidates ($T_{\text{gemini}}, T_{\text{gpt}}$) that passed the numerical check, we applied the Back-Translation Quality Selection (BTQS) methodology.

1. **Back-Translation**: Each candidate translation is back-translated into English (E') using the model used for its generation.

2. **Semantic Similarity Measurement**: To evaluate semantic equivalence between the source (E) and back-translated text (E'), cosine similarity is calculated using the `BAAI/bge-m3` encoder.

$$s(T) = \cos\left(\phi(E), \phi(E')\right) \quad (2)$$

3. **Final Selection**: The translation with the higher similarity score is adopted as the final data.

$$T^* = \underset{T \in \{T_{\text{gemini}}, T_{\text{gpt}}\}}{\arg\max} \; s(T) \quad (3)$$

Target Lang	Acc (%)	Δ	Target Lang	Acc (%)	Δ
English (Baseline)	**90.00**	–	Quechua (qu)	89.60	-0.40
Tamil (ta)	93.60	+3.60	Guarani (gn)	90.00	0.00
Gujarati (gu)	91.60	+1.60	Basque (eu)	90.00	0.00
Kannada (kn)	91.20	+1.20	Amharic (am)	90.00	0.00
Khmer (km)	91.20	+1.20	Javanese (jv)	90.00	0.00
Sinhala (si)	91.20	+1.20	Uyghur (ug)	89.60	-0.40
Uzbek (uz)	91.20	+1.20	Tajik (tg)	89.60	-0.40
Yoruba (yo)	90.80	+0.80	Sundanese (su)	89.60	-0.40
Kyrgyz (ky)	90.80	+0.80	Burmese (my)	89.60	-0.40
Sindhi (sd)	90.40	+0.40	Georgian (ka)	89.60	-0.40
Nepali (ne)	90.40	+0.40	Pashto (ps)	89.60	-0.40
Punjabi (pa)	90.40	+0.40	Armenian (hy)	89.60	-0.40
Maltese (mt)	90.40	+0.40	Malayalam (ml)	89.60	-0.40
Lao (lo)	90.40	+0.40	Malagasy (mg)	89.60	-0.40
Somali (so)	90.00	0.00	Mongolian (mn)	89.20	-0.80
Odia (or)	90.00	0.00	Kazakh (kk)	89.20	-0.80
Marathi (mr)	90.00	0.00	Kurdish (ku)	89.20	-0.80
Cebuano (ceb)	90.00	0.00			

Table 5: Reasoning accuracy of Qwen2.5-7B on MGSM tasks using Round-trip Translation (English → Pivot → English). We used the target languages as pivots to validate the semantic consistency of the translated dataset. Δ denotes the performance difference relative to the English baseline.

C.3 Quality Control and Verification Results

To confirm the reliability of the constructed dataset, we doubly verified linguistic accuracy and logical integrity based on the outputs of the BTQS pipeline.

First, samples with BTQS similarity scores below the threshold (0.8) from the previous stage were considered defective and primarily excluded. Instead of simple exclusion, a **Regeneration** loop was immediately executed for these samples to iteratively correct them until all data met the quality standards.

For the finally selected samples, **Round-trip Reasoning Validation** was performed to check the preservation of mathematical essence. To maximize efficiency, this was conducted by **reusing the back-translated text generated during the BTQS evaluation process** without additional translation. That is, English text generated via 34 target languages as pivots was input into the reasoning model (QWEN2.5-7B), and logical consistency was verified by evaluating whether the model correctly derived the answer.

The experimental results in Tab. 5 strongly support the utility of this verification pipeline. Data routed through many pivot languages, including Tamil (93.6%), recorded reasoning accuracies comparable to or even exceeding the baseline (90.0%). Even Kazakh, which showed relatively lower performance, maintained a decent accuracy of 89.2%, suggesting that information and logical structures essential for problem-solving were robustly pre-

served without loss during the round-trip conversion to the target language.

Of course, this round-trip reasoning accuracy does not perfectly guarantee the grammatical completeness of the target language text itself. The possibility that the high-performance reasoning model self-corrected minor noise in the back-translated text to derive the answer cannot be ruled out. However, in low-resource language environments where securing native evaluators is realistically difficult, this is judged to be the most effective and scalable proxy metric for determining logical information loss.

Detailed statistics of the finally constructed dataset are shown in Tab. 6. The dataset consists of a total of 8,500 high-quality samples, with final contributions by model being 62.1% for Gemini and 37.9% for GPT.

Statistic	**Value**
Total Target Languages	34
Samples per Language	250
Total Samples	8,500
Gemini Selection Ratio	62.1%
GPT Selection Ratio	37.9%
Avg. BT Similarity Score	0.96

Table 6: Final statistics of the MGSM-LowResource dataset.

Code	Language	Family	Morph.	Order
		Target languages		
kk	Kazakh	Turkic	Agg	SOV
ky	Kyrgyz	Turkic	Agg	SOV
mn	Mongolian	Mongolic	Agg	SOV
ug	Uyghur	Turkic	Agg	SOV
hy	Armenian	IE (Armenian)	Agg†	SOV
lo	Lao	Kra–Dai	Ana	SVO
eu	Basque	Isolate	Agg	SOV
mt	Maltese	Afro-Asiatic (Semitic)	Fus	SVO
am	Amharic	Afro-Asiatic (Semitic)	Fus	SOV
ne	Nepali	IE (Indo-Aryan)	Fus	SOV
si	Sinhala	IE (Indo-Aryan)	Fus	SOV
ps	Pashto	IE (Iranian)	Fus	SOV
sd	Sindhi	IE (Indo-Aryan)	Fus	SOV
pa	Punjabi	IE (Indo-Aryan)	Fus	SOV
gu	Gujarati	IE (Indo-Aryan)	Fus	SOV
mr	Marathi	IE (Indo-Aryan)	Fus	SOV
or	Odia	IE (Indo-Aryan)	Fus	SOV
ta	Tamil	Dravidian	Agg	SOV
ml	Malayalam	Dravidian	Agg	SOV
		Skeleton languages		
en	English	IE (Germanic)	Ana/Fus	SVO
zh	Chinese	Sino-Tibetan	Ana	SVO
es	Spanish	IE (Romance)	Fus	SVO
ru	Russian	IE (Slavic)	Fus	SVO
ko	Korean	Koreanic	Agg	SOV
th	Thai	Kra–Dai	Ana	SVO

Table 7: Typological features of the 19 target and six skeleton languages. **Morph.**: dominant morphological type (Ana: analytic/isolating; Agg: agglutinative; Fus: fusional). **Order**: basic constituent order. Scripts are listed in Tab. 4. †Armenian exhibits agglutinative nominal morphology. These features derive surface affinity and structural alignment as defined above.

D Linguistic Typology and Affinity/Alignment Classification

To make the notions of *surface affinity* and *structural alignment* used in Sec. 4.5 precise and reproducible, we define them operationally as functions of language-level typological features, listed for all target and skeleton languages in Tab. 7.

Definitions. For a target language t and a skeleton language s:

- **Surface affinity** $\mathrm{Aff}(t, s)$=1 iff t and s share at least one of: (i) genetic language family, (ii) writing script, or (iii) geographic macro-region; otherwise 0.
- **Structural alignment** $\mathrm{Align}(t, s)$=1 iff t and s share *both* (i) dominant morphological type (analytic, agglutinative, or fusional) and (ii) basic constituent order (e.g., SOV vs. SVO); otherwise 0.

These are *descriptive* categories, not predictors. As shown in Sec. 4.5, structural alignment does not guarantee positive transfer (e.g., Maltese is alignment-positive yet evaluation-sensitive), and surface affinity does not prevent negative transfer (e.g., ug–zh and mn–ru share affinity but are misaligned and transfer negatively). The classification removes ambiguity from our terminology rather than asserting a causal rule.

Worked examples. Under these definitions, lo–th is both affine (Kra–Dai family) and aligned (analytic, SVO); eu–ko is *not* affine but is aligned (agglutinative, SOV); ug–zh and mn–ru are affine (geography or script) but *not* aligned. This matches the patterns reported in Sec. 4.5.

E Computational Cost Analysis

Fig. 4, 5 presents the quantitative analysis of the change in computational cost due to the introduction of skeletons. The average length of the skeleton is highly concise, with an overall mean of 120.42 tokens. The average total token count (Skeleton + RS) is 1071.75, representing an increase of approximately 146.68 tokens compared to the vanilla baseline mean of 925.07. Notably, even in languages with high token consumption such as Swahili (sw),

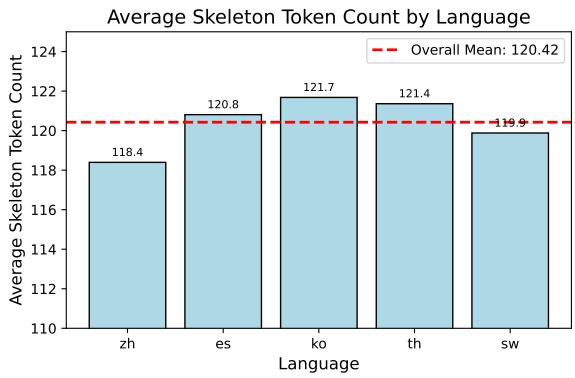


Figure 4: Average token length of skeletons across query languages.

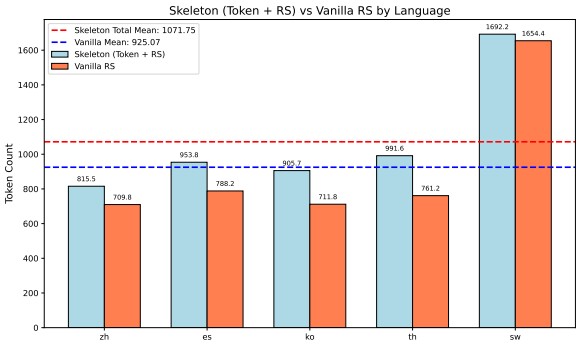


Figure 5: Average generated token counts of the Qwen-series models.

which requires 1692.2 tokens for the full process, the relative overhead remains manageable. These results suggest that the skeleton efficiently guides the model's reasoning process, suppressing unnecessary elaboration or inefficient operations during generation.

F Qualitative Analysis

F.1 Quality Assessment of English Skeletons

Figure 6: Quality evaluation results for five models in the PolyMath

Fig. 6 (Left) shows the change in skeleton generation quality according to model scale. A distinct trend of improved skeleton quality with larger

Figure 7: Skeleton quality heatmap across language pairs. English skeletons yield highest quality (7.57), while Thai performs worst (5.65).

model scale was observed; in particular, large-scale models of 70B or larger recorded a high quality score of 6.7 on average.

Furthermore, the correlation analysis results in Fig. 6 (Right) shed light on the relationship between skeleton quality factors and final reasoning performance. 'Coherence' ($r = 0.82$) and 'Completeness' ($r = 0.74$) showed strong positive correlations with final performance, whereas 'Specificity' ($r = 0.40$) showed a relatively lower correlation. This implies that for successful reasoning in the target language, it is more important for the skeleton to include the logical flow of problem-solving and key elements without omission, rather than describing detailed steps specifically.

F.2 Cross-Lingual Skeleton Quality and the Quality–Performance Gap

The translation ablation in Sec. 5 controls skeleton *content* to isolate surface form. Here we provide complementary evidence from the opposite direction: we score the *intrinsic generation quality* of skeletons across languages using an LLM-as-a-Judge, and ask whether quality alone can explain the cross-lingual effects in Sec. 4.5.

English dominates intrinsic quality. As shown in Fig. 7, English attains the highest average skeleton-quality score (7.57), followed by Spanish (7.35), Chinese (7.24), and Russian (7.13), while Korean (6.69) and Thai (5.65) score lower. This is consistent with English serving as a stable default skeleton language for most targets (Sec. 4.5).

Quality does not track performance under structural alignment. For several structurally aligned pairs, a *lower*-quality skeleton nonetheless yields

higher performance. For Lao, the Thai skeleton scores 1.29 points below English in quality (6.48 vs. 7.77) yet improves accuracy by +2.81; for Basque, the Korean skeleton scores 1.19 points below English (6.18 vs. 7.37) yet improves accuracy by +6.32. A quality-only account would incorrectly predict English to be optimal here.

No gap in the absence of structural alignment. Conversely, where the target and skeleton share surface affinity but *not* structural alignment, comparable quality coincides with negative transfer. For Uyghur, the Chinese skeleton has quality comparable to English (7.40 vs. 7.54) but performance drops by −6.45; for Mongolian, the Russian skeleton even slightly exceeds English in quality (7.22 vs. 7.19) yet performance drops by −5.00. Following the affinity/alignment distinction in Sec. 4.5, these are precisely the pairs that share surface affinity (script or geography) but differ in morphological type and word order.

Takeaway. Together with the translation ablation (Sec. 5), this indicates that skeleton-language effects are not reducible to generation quality: structural alignment can compensate for lower intrinsic quality, whereas its absence is not offset by high quality. We regard this as supporting evidence for treating skeleton language as a design variable to be identified through exploration, and—consistent with our framing throughout—as an exploratory hypothesis rather than a causal claim.

F.3 Error Analysis: Impact of Skeleton

Language	Method	Problem	Conceptual	Reasoning	Calculation	Output Gen.
kk	CoT-ℓ_t	100	61	129	33	116
	+SKELETON	91	54	105	18	135
ug	CoT-ℓ_t	106	74	156	57	95
	+SKELETON	100	60	134	33	135
mn	CoT-ℓ_t	107	78	157	47	88
	+SKELETON	108	62	142	41	115
hy	CoT-ℓ_t	80	57	108	31	155
	+SKELETON	88	50	95	22	154
lo	CoT-ℓ_t	115	82	159	35	179
	+SKELETON	133	75	156	27	123
eu	CoT-ℓ_t	135	88	160	47	76
	+SKELETON	140	85	155	39	97
mt	CoT-ℓ_t	112	83	154	47	98
	+SKELETON	124	66	154	48	119
ne	CoT-ℓ_t	77	34	93	23	135
	+SKELETON	72	44	86	26	139

Table 8: Comparison of CoT-ℓ_t and +SKELETON Results Across Different Reasoning Stages

To investigate the underlying causes of the performance changes observed in Tab. 2, we conducted an in-depth error analysis on the entire evaluation dataset using the LLM-as-a-Judge methodology. Tab. 8 summarizes the frequency changes of the five key error types before and after applying the skeleton. The detailed evaluation protocol is described in Appendix B.2.

To qualitatively analyze the effectiveness of the skeleton, we constructed an automated error classification pipeline using the `gpt-5-mini` model. For each incorrect sample, the model response was provided as input, and the prompt was designed to identify the most decisive error type among the following five categories:

1. **Problem Comprehension Error**: Misunderstanding the problem's goal, constraints, or key variables.
2. **Conceptual Error**: Incorrect application or omission of necessary formulas, theorems, or scientific principles.
3. **Reasoning Error**: Errors in logical flow or inclusion of incorrect reasoning steps.
4. **Calculation Error**: Mistakes in arithmetic or symbolic operations despite a correct approach.
5. **Output Generation Error**: Format errors (missing `\boxed{}`), infinite repetition, language mismatch, etc.

The evaluation model outputs the presence of each error type and a corresponding description in JSON format, which allowed for a quantitative comparative analysis of error patterns by language and condition.

Skeletons as Cognitive Scaffolding Analysis of language groups with significant performance improvement (kk, ug, mn, hy) confirmed that the introduction of skeletons significantly suppresses Reasoning Errors and Calculation Errors. In the case of Kazakh (kk), which recorded the largest performance gain, Reasoning Errors decreased from 129 to 105 (−18.6%) and Calculation Errors plummeted from 33 to 18 (−45.5%) after applying the skeleton. This suggests that the English skeleton serves as *cognitive scaffolding* that decomposes complex problems into manageable sub-units, effectively preventing the model from deviating from the logical path or committing simple arithmetic mistakes during the reasoning process.

Semantic Interference Causing Comprehension Failure Conversely, in language groups where

Method	MGSM							MATH-500							PolyMath						
	zh	es	ko	th	sw	te	AVG.	zh	es	ko	th	sw	te	AVG.	zh	es	ko	th	sw	te	AVG.
Llama-3.1-8B-Instruct																					
CoT(ℓ_q = target)	71.5	74.7	58.6	65.5	61.7	56.2	64.7	33.2	37.7	29.8	31.0	31.9	25.1	31.5	23.6	24.9	26.2	19.3	27.6	16.2	23.0
+ SKELETON	68.3	75.5	61.0	65.1	63.8	57.4	65.2	34.7	35.6	34.0	32.0	34.2	25.9	32.7	24.1	26.3	24.4	21.1	29.1	16.8	23.6
CoT(ℓ_q = EN†)	69.3	78.3	64.9	69.1	68.3	63.9	69.0	33.8	41.6	29.7	33.3	32.5	26.3	32.9	22.4	27.6	24.4	21.6	32.4	17.3	24.3
+ SKELETON	69.8	73.9	63.7	70.3	66.7	62.7	67.9	37.7	37.3	32.9	31.6	38.8	29.6	34.7	26.3	28.0	27.3	22.4	33.9	17.3	25.9
Llama-3.1-70B-Instruct																					
CoT(ℓ_q = target)	86.3	88.0	83.9	86.8	83.9	81.9	85.1	50.1	50.1	49.0	51.0	48.6	43.2	48.7	29.8	28.6	32.2	29.1	35.0	25.0	30.0
+ SKELETON	88.8	81.5	82.7	87.2	78.9	80.3	83.2	50.3	45.6	50.4	52.3	49.1	45.4	48.9	30.7	29.2	31.2	30.4	33.8	26.9	30.4
CoT(ℓ_q = EN†)	85.5	88.4	84.7	85.1	83.3	83.1	85.0	49.5	50.9	51.9	47.2	51.1	46.2	49.5	29.4	32.9	32.9	28.7	36.3	26.0	31.0
+ SKELETON	85.9	82.3	83.9	82.7	82.5	82.7	83.3	51.1	50.9	51.2	49.0	48.2	45.6	49.3	29.2	29.7	31.7	28.3	36.3	27.3	30.4

Table 9: Multilingual mathematical reasoning results for the Llama 3.1 series (8B, 70B). The experiments use the MGSM, MATH-500, and PolyMath benchmarks. ℓ_q = target indicates that questions are presented in the target language, while ℓ_q = EN† denotes the use of English-translated questions. The skeleton language is fixed to English in all cases (ℓ_s = EN). 'AVG.' denotes the average over the six languages.

performance improvement was marginal or declined (lo, eu, mt, ne), an increase in Problem Comprehension Errors acted as the main failure factor. For Lao (lo), while the change in Reasoning Errors was minimal, Problem Comprehension Errors tended to increase from 115 to 133 (+15.7%). This is interpreted as a phenomenon where the English skeleton distorted the core requirements of the original problem due to semantic interference between languages, or the model became overly biased towards the English instructions of the skeleton, missing the context of the original problem.

G Llama 3.1 Experimental Results

While the main text presented experimental results centering on the QWEN2.5 series, we performed the same experiments on the Meta LLAMA 3.1 series (8B, 70B) to verify the generalizability of the methodology.

Tab. 9 summarizes the results of the MGSM, MATH-500, and PolyMath benchmarks for the LLAMA 3.1 model. The overall trends appeared similar to the QWEN2.5 series:

- **Improvement in Low-Resource Languages**: In the CoT-$en^\dagger$ setting of the LLAMA-3.1-8B model, a performance improvement from 32.5 to 38.8 (+6.3) was observed in MATH-500 for Swahili (sw) when skeletons were applied.
- **Diminishing Returns with Scale**: In the LLAMA-3.1-70B model, the effect of skeletons was minimal, or performance degradation occurred in some languages (es: 88.0 → 81.5).
- **Differences by Benchmark Difficulty**: The most consistent improvement effects appeared in MATH-500, while MGSM and PolyMath showed large variations by language.

Averaged over the six languages, Llama-3.1-8B shows a small positive skeleton effect, whereas the effect largely disappears or reverses for Llama-3.1-70B (Tab. 9): the overall mean skeleton effect is $+0.81$ pp (8B, target-language queries) and $+0.75$ pp (8B, English-translated queries), versus -0.43 and -0.81 pp for 70B. The skeleton effect therefore varies across model variants and evaluation conditions rather than being uniformly positive; what is shared with Qwen2.5 is the qualitative tendency of the effect to shrink as model size and baseline performance increase (see also Appendix H). However, since Llama 3.1 has a relatively lower proportion of multilingual data, the baseline performance in low-resource languages was measured lower than that of Qwen2.5. Particularly for Llama-3.1-70B, which already demonstrated high baseline performance in most conditions, the room for additional improvement by skeletons was limited.

H Additional Model Family: Ministral

Model	MGSM	MATH-500	PolyMath	AVG.	AVG.*
Ministral-3B	+1.34	+1.79	+0.74	+1.29	+0.38
Ministral-14B	−1.10	+0.35	+1.08	+0.11	−0.12

Table 10: English-skeleton effect Δ (pp) for the Ministral family under the CoT-ℓ_t setting, averaged over the five languages of Tab. 1 (zh, es, ko, th, sw) per benchmark. 'AVG.' denotes the average over all five languages; AVG.* excludes Swahili (sw), which retains relatively few valid samples.

To examine whether the scale-dependent pattern is specific to Qwen2.5 and Llama 3.1, we additionally evaluate a third model family, MINISTRAL-INSTRUCT (3B and 14B) (Liu et al., 2026). To keep the comparison consistent, we fix the query condition to the target language (CoT-ℓ_t) and eval-

uate both models on the three benchmarks and five languages of Tab. 1, yielding 30 cells. Δ denotes the change in accuracy (pp) from adding an English skeleton under the same query condition.

As shown in Tab. 10, Ministral-3B improves by $+1.29$ pp on average, whereas the average effect for Ministral-14B is only $+0.11$ pp; excluding Swahili (sw), which retains relatively few valid samples, the averages are $+0.38$ and -0.12 pp, respectively. Across the 30 cells, baseline accuracy is negatively correlated with the skeleton effect (Spearman $\rho = -0.568$, nominal $p = 0.0011$). Thus, the English-skeleton effect is not consistently positive in this additional family either, and it varies across model variants and benchmarks. Because we evaluate only two sizes per family, we do not interpret this as a general scaling law; rather, across all three families we observe the same qualitative tendency that the average skeleton effect decreases as model size and baseline performance increase (mean Δ under CoT-ℓ_t: Qwen2.5 $+1.73 \rightarrow +1.00$ from 7B to 72B; Llama-3.1 $+0.81 \rightarrow -0.43$ from 8B to 70B; Ministral $+1.29 \rightarrow +0.11$ from 3B to 14B). These results distinguish the *generality of the framework*—the language decomposition (ℓ_q, ℓ_s, ℓ_a) applies to any model—from the *generality of the observed performance effects*, which do not automatically transfer across models.

I PolyMath Difficulty-Level Analysis

Model	CoT-ℓ_t		CoT-$en^\dagger$	
	Base	+SKELETON	Base	+SKELETON
QWEN2.5-7B	4.75	4.91 (+0.16)	5.12	5.82 (+0.70)
QWEN2.5-14B	6.62	8.40 (+1.78)	7.13	7.56 (+0.43)
QWEN2.5-72B	9.27	9.97 (+0.70)	8.92	8.96 (+0.04)

Table 11: Skeleton gains on the two hardest PolyMath subsets (Top + High; 250 of 500 problems), as Exact-Match accuracy (%) averaged over five languages (zh, es, ko, th, sw). Parentheses show the gain (Δ) from adding the English skeleton. Compared with the full-set gains (Tab. 1), improvements shrink on the harder subset, consistent with the diminishing effect of skeletons as difficulty increases.

PolyMath is organized into four difficulty levels (Low, Medium, High, Top; 125 problems each). To probe the effect of problem difficulty more directly than the full-set average, we restrict evaluation to the two hardest subsets (Top + High; 250 problems) and recompute the English-skeleton gain for the QWEN2.5 family (7B, 14B, 72B), averaged over five languages (zh, es, ko, th, sw). Tab. 11 reports the results.

Compared with the gains aggregated over all 500 problems, improvements on the harder subset generally shrink: for QWEN2.5-7B the average CoT-ℓ_t gain drops from $+1.20$ to $+0.16$, and for QWEN2.5-72B from $+0.90$ to $+0.70$. The mid-sized QWEN2.5-14B is an exception, showing a larger gain ($+1.78$) on the harder subset, suggesting that skeleton guidance can still aid mid-sized models on difficult problems. Overall, the tendency is consistent with our main finding that the benefit of skeletons diminishes as problem difficulty increases.

J Statistical Significance Analysis

Target	Skeleton	Δ (%)	p
Positive effects (Non-English skeleton > English skeleton)			
mt	ru	$+8.62$	0.001
mt	th	$+6.90$	0.008
mt	es	$+6.61$	0.041
Negative effects (English skeleton > Non-English skeleton)			
ug	th	-10.25	0.002
mn	th	-8.47	0.016
kk	ko	-7.66	0.030

Table 12: Pairs that reached nominal significance in McNemar's test before correction for multiple comparisons ($p < 0.05$). Δ denotes the accuracy difference between the non-English skeleton and the English skeleton.

To assess the statistical reliability of the non-English skeleton effects discussed in Sec. 4.5, we conducted McNemar's tests. McNemar's test compares paired binary outcomes over the same set of instances, focusing on the transition patterns between correct and incorrect predictions under two conditions. In our analysis, we tested whether item-level correctness differed significantly between the English skeleton and each non-English skeleton. Using the greedy decoding results of QWEN2.5-7B, we examined a total of 95 target–skeleton pairs, corresponding to 19 low-resource target languages and five non-English skeleton languages.

Results. Among the 95 pairs, six reached nominal significance before correction for multiple comparisons ($p < 0.05$). These include both positive effects, where a non-English skeleton outperformed the English skeleton, and negative effects, where the English skeleton outperformed the non-English skeleton.

The three positive effects were all observed for Maltese (mt), with mt–ru showing the strongest signal ($p = 0.001$). In contrast, negative effects were

observed for Uyghur–Thai, Mongolian–Thai, and Kazakh–Korean. These results indicate that non-English skeletons can also significantly degrade performance relative to English skeletons in some cases.

The pairs discussed in the main text as directionally consistent cross-lingual effects, namely eu–ko ($p = 0.072$), hy–ko ($p = 0.361$), and lo–th ($p = 0.401$), do not reach nominal significance under McNemar's test. We therefore do not interpret them as effects confirmed by a single statistical test. Instead, we treat them as consistency-based observations, since the direction of the effect remains stable across greedy decoding and multi-rollout evaluation. In other words, McNemar's test examines item-level correctness transitions under a single decoding setting, whereas our robustness analysis evaluates whether the direction of the effect is preserved across evaluation protocols.

For Maltese, the positive effects are nominally significant under greedy decoding, but they are substantially attenuated or even reversed under multi-rollout evaluation. This is consistent with our interpretation in the main text that Maltese represents an evaluation-sensitive case rather than a directionally consistent transfer pattern; the cross-benchmark results in Appendix K further show that the Maltese gains are benchmark-dependent.

English Skeleton vs. No Skeleton. Following the same procedure, we also tested whether English skeletons significantly outperform the no-skeleton CoT baseline on the 19 low-resource MGSM languages (QWEN2.5-7B, CoT-ℓ_t). The effect was positive for 13 of the 19 languages, with a mean language-level change of $+2.38$ pp. After FDR correction, however, significant improvements remained for only two languages: Kazakh (kk; $40.08 \rightarrow 51.42$, $\Delta = +11.34$, $p_{\text{FDR}} = 0.014$) and Uyghur (ug; $27.13 \rightarrow 37.65$, $\Delta = +10.52$, $p_{\text{FDR}} = 0.014$). The -5.65 pp decrease for Maltese was nominally significant before correction ($p = 0.049$) but not after FDR correction ($p_{\text{FDR}} = 0.307$), and no other language showed a significant difference after correction. When the 19 languages are pooled at the item level ($n = 4{,}628$), the overall effect is significant ($\Delta = +2.27$ pp, $p = 0.0014$). In other words, the benefit of English skeletons is real but too small to detect at the individual-language level, and it is driven by a few languages such as kk and ug.

K Cross-Benchmark Generalization on MSVAMP

Target	Skel.	MGSM Δ	MSVAMP Δ	Outcome
hy	ko	+5.17	+2.75	Retained
lo	zh	+3.63	+2.63	Retained
lo	th	+2.81	+2.42	Retained
eu	ko	+6.32	0.00	Disappeared
mt	es	+6.61	+2.00	Attenuated
mt	th	+6.90	+0.43	Attenuated
mt	ru	+8.62	−1.27	Reversed
ug	th	−10.25	−2.14	Retained (neg.)
mn	th	−8.47	−2.53	Retained (neg.)
kk	ko	−7.66	−4.63	Retained (neg.)

Table 13: Cross-benchmark transfer of skeleton-language effects from MGSM to MSVAMP (QWEN2.5-7B, CoT-ℓ_t). Δ: accuracy difference (pp) between the non-English and the English skeleton. Pairs were fixed using only the MGSM results before inspecting MSVAMP. Across all 35 target–skeleton cells, the sign is preserved in 23/35 (66%).

To test whether skeleton-language effects observed on MGSM generalize to an unseen test set, we evaluate on MSVAMP (Chen et al., 2024), a separate benchmark whose problems do not overlap with MGSM (up to 1,000 mathematical reasoning problems per language). Since MSVAMP natively supports only ten languages, we obtained additional low-resource target languages using the same translation and verification pipeline described in Appendix C, extending it to eight languages (ko, eu, hy, lo, mn, ug, mt, and kk). We use the seven target languages for which MGSM-LowResource results are available (eu, hy, lo, mn, ug, mt, and kk).

Protocol. To rule out post-hoc selection, the evaluation targets, selection rule, and skeleton candidates were fixed using only the MGSM results before inspecting MSVAMP. We selected eu, hy, and lo (directionally consistent pairs in Sec. 4.5), mn and ug (asymmetric-negative cases), and mt and kk (nominally significant results in Appendix J). For each target language, we applied the five non-English skeleton languages (zh, es, ru, ko, and th) under the target-language query condition and computed Δ, the accuracy difference relative to the English skeleton, on both benchmarks, yielding $7 \times 5 = 35$ target–skeleton cells.

Results. Across the 35 cells, effect sizes on MGSM and MSVAMP show a moderate positive rank correlation (Spearman $\rho = 0.383$, nominal $p = 0.023$), and the sign of the effect is preserved in 23 of the 35 cells (66%; binomial $p = 0.090$).

Tab. 13 summarizes the key pairs. Two observations stand out. First, the asymmetric-negative pattern transfers relatively well: among the 15 cells for mn, ug, and kk, 14 remain negative on MSVAMP, with only ug–ru reversing ($-4.11 \rightarrow +1.79$). Second, the positive Maltese effects are not robust: mt–es remains positive ($+6.61 \rightarrow +2.00$), but mt–th largely diminishes ($+6.90 \rightarrow +0.43$) and mt–ru reverses ($+8.62 \rightarrow -1.27$). The transfer of positive selections is mixed: hy–ko, lo–zh, and lo–th retain their positive direction, whereas eu–ko vanishes to 0.00.

Implications. We therefore do not claim that the optimal skeleton language can be specified in advance for an arbitrary target language. Within the evaluated subset, results from an existing benchmark are more useful for identifying potentially risky skeleton candidates and prioritizing subsequent evaluations than for directly determining an optimal skeleton language. This asymmetry is an exploratory observation, and separate validation remains necessary for a new benchmark, model, or task. As MGSM and MSVAMP focus on elementary arithmetic, testing transfer on harder benchmarks is left for future work.

L Model Performance Across Other Low-Resource Languages

L.1 Results on Telugu

Tab. 14 presents the performance changes when applying the skeleton to Telugu.

L.2 Results on All Low-Resource Languages

Although our experiments cover 34 languages in total, target-language CoT (CoT-ℓ_t) fails to operate reliably for a subset of languages, making fair comparison difficult. We therefore focus our main analysis on 19 languages where target-language reasoning is stable. For completeness and transparency, results for all 34 languages are reported in Fig. 8.

Method	MATH-500 (te)	PolyMath (te)	MGSM (te)
Qwen2.5-7B-Instruct			
CoT-ℓ_t	29.8	12.2	30.40
+SKELETON	27.8	12.4	33.60
CoT-$en^{\dagger}$	35.7	17.3	60.40
+SKELETON	33.9	17.7	54.00
Qwen2.5-14B-Instruct			
CoT-ℓ_t	43.4	18.6	54.80
+SKELETON	45.9	19.8	55.20
CoT-$en^{\dagger}$	45.1	24.2	71.89
+SKELETON	43.0	22.2	73.90
Qwen2.5-72B-Instruct			
CoT-ℓ_t	53.2	23.7	65.20
+SKELETON	55.4	27.4	68.00
CoT-$en^{\dagger}$	52.4	28.2	78.40
+SKELETON	55.2	28.8	82.80
Llama-3.1-8B-Instruct			
CoT-ℓ_t	25.8	16.2	56.40
+SKELETON	25.0	16.8	57.60
CoT-$en^{\dagger}$	26.7	17.3	63.60
+SKELETON	30.1	17.3	62.80
Llama-3.1-70B-Instruct			
CoT-ℓ_t	43.6	25.0	82.00
+SKELETON	45.9	26.9	80.40
CoT-$en^{\dagger}$	46.7	26.0	83.20
+SKELETON	46.1	27.3	82.80

Table 14: Telugu (te) Accuracy (%) on MATH-500, PolyMath, and MGSM

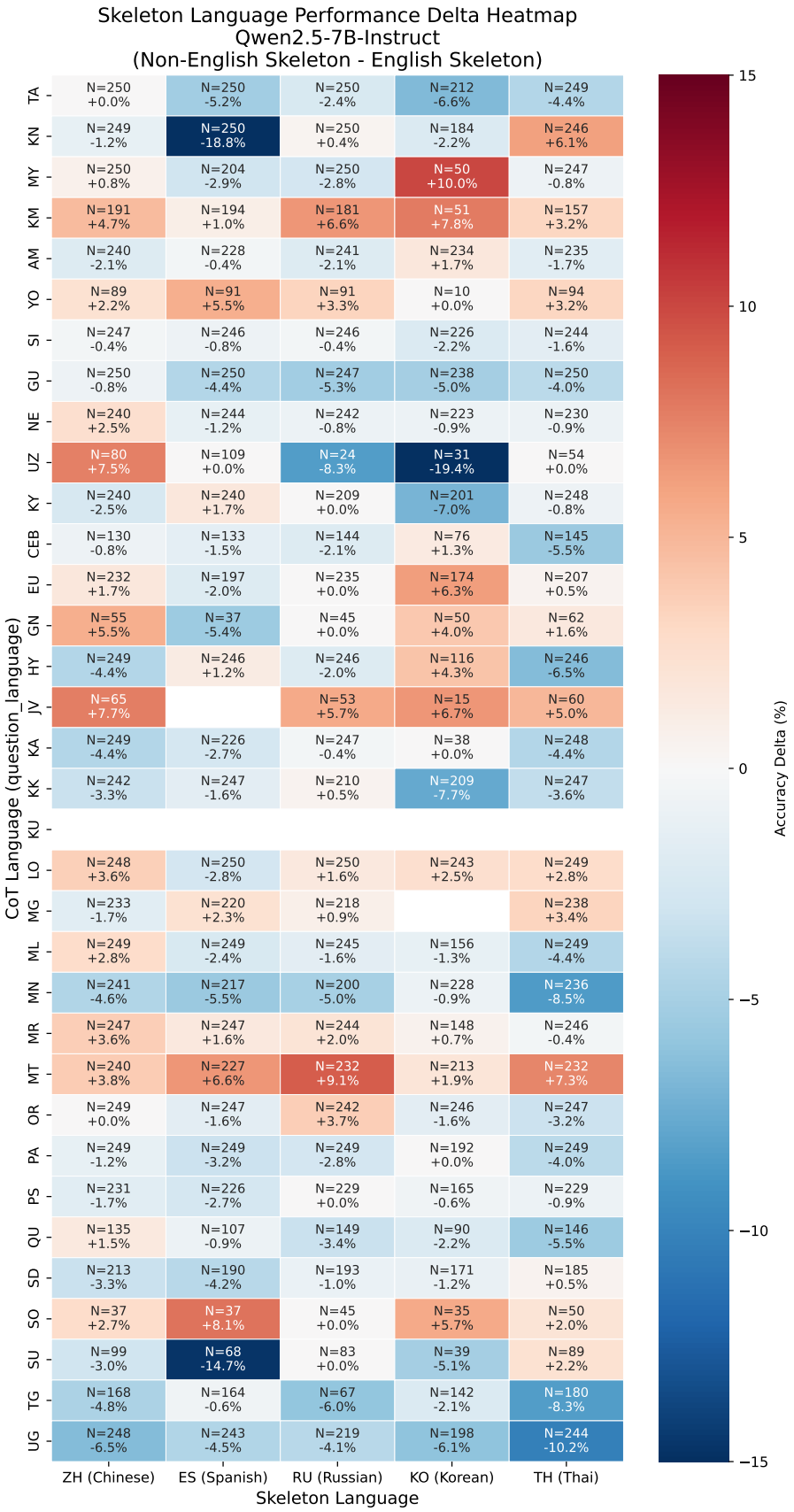


Figure 8: Performance differences (Non-English English) under the CoT-ℓ_t setting using QWEN2.5-7B across all 34 target languages. While English generally serves as a stable default skeleton language, several target languages exhibit gains with specific Non-English skeletons, indicating that skeleton effectiveness is language-dependent.