Language as Reasoning Path: Uncovering Language-Specific Mathematical Reasoning in Large Language Models

Anonymous ACL submission

Abstract

Do large language models reason the same way regardless of the language they are prompted in? We investigate this question through a large-scale multilingual mathematical reasoning study. We construct a dataset of 6,000 competition-level math problems from DAPO, each translated into 10 typologically diverse languages, and collect 60,000 step-by-step solutions from DeepSeek-V4-Pro. All non-English responses are translated into English, yielding 54,000 cross-language-comparable reasoning traces. Our analysis reveals three key findings. First, while English achieves the highest overall accuracy (67.5%), it is *not* the most token-efficient reasoning language: Chinese, German, Korean, and French each produce the shortest correct solution more often than English. Second, 3,525 out of 6,000 problems (58.8%) exhibit at least two distinct correct reasoning paths across languages, demonstrating that language systematically diversifies the model's solution strategies. Third, we identify a "weak-language advantage" phenomenon: on 192 problems, at least one non-stronger language answers correctly while all five stronger languages fail. These results reframe multilingual prompting not merely as a robustness test, but as a *reasoning-path search mechanism* that exposes capabilities invisible under English-only evaluation. We release the full dataset, translated traces, and analysis code.

1 Introduction

If language were merely a surface wrapper around a universal reasoning engine, then translating a math problem from English to Korean should produce the same solution in different words. But what if it produces a different solution entirely?

Large language models have become increasingly capable mathematical reasoners. With chain-of-thought prompting (Wei et al., 2022), self-consistency decoding (Wang et al., 2023), and reinforcement learning from verifiable rewards, modern models can solve multi-step competition problems at levels that would have been unimaginable a few years ago. Yet the methodological default across virtually all of this progress is English-centric: researchers elicit, evaluate, and analyze reasoning in English, implicitly treating English as the optimal—or at least neutral—reasoning interface.

This default hides a fundamental question. The language of a prompt can affect lexical priors, memorized solution templates, decomposition habits, and the distribution over intermediate representations. When two prompts are faithful translations of the same mathematical problem, they may nonetheless steer the model toward different tools, different intermediate objects, and entirely different solution strategies. In this view, language is not a neutral carrier but a *reasoning control signal* that steers the model into different regions of its solution space.

We study this phenomenon at scale. We construct a multilingual mathematical reasoning dataset from 6,000 DAPO problems (Yu et al., 2025), each translated into 10 typologically diverse languages spanning six language families, multiple writing systems, and a wide range of LLM resource levels. We collect step-by-step solutions from DeepSeek-V4-Pro, a state-of-the-art open-source reasoning model, and translate all non-English responses into English for cross-language comparison. This design enables us to ask not just *whether* accuracy varies across languages—it does—but *how* the reasoning paths themselves differ.

Our analysis is organized around three research questions:

RQ1. *How does mathematical reasoning accuracy and efficiency vary across prompt languages?*

RQ2. *Do different languages induce genuinely distinct reasoning paths on the same problem?*

RQ3. *Can non-stronger languages reveal correct*

solutions that stronger languages miss?

The answers are striking. English leads in accuracy (67.5%) but Chinese, German, Korean, and French each produce the shortest correct answer more often. Among the 5,297 solvable problems, 3,525 (66.5% of solvable, 58.8% of all) exhibit multiple distinct correct reasoning-path groups across languages. And on 192 problems, at least one non-stronger language succeeds where all five stronger languages fail—a phenomenon we term the *weak-language advantage* (Figure 1).

Taken together, these results position multilingual prompting not as a robustness diagnostic but as an active search over solution strategies—a tool for discovering mathematical capabilities that remain hidden under English-only evaluation. This perspective has direct implications for reasoning ensembles, evaluation protocols, and our understanding of how language interacts with learned mathematical knowledge.

Contributions. Our paper makes three contributions:

1. We construct and release a large-scale multilingual mathematical reasoning dataset: 6,000 problems $\times$ 10 languages = 60,000 solutions with correctness labels, token statistics, reasoning-path annotations, and 54,000 English-translated traces.
2. We demonstrate that prompt language systematically diversifies reasoning paths: 58.8% of problems exhibit multiple distinct correct solution strategies across languages, with consistent domain-specific method preferences.
3. We identify and characterize the weak-language advantage: non-stronger languages can solve problems that all stronger languages miss, revealing mathematical capabilities that are language-specific rather than language-invariant.

2 Related Work

Multilingual reasoning in LLMs. Cross-lingual transfer and multilingual evaluation have been studied extensively (Conneau et al., 2020; Pires et al., 2019; Xue et al., 2021). Recent work shows that LLM reasoning degrades in non-English languages (Shi et al., 2023; Lai et al., 2023; Ahuja et al., 2023), motivating translated benchmarks and cross-lingual prompting strategies (Huang et al., 2023; Qin et al., 2023). Chen et al. (2024) and Anand et al. (2024) study multilingual math specifically. Our work shifts the question: rather than quantifying how much non-English performance degrades, we ask whether non-English prompts expose *different* reasoning strategies entirely.

Language-specific representations in LLMs. Evidence is accumulating that multilingual LLMs maintain language-specific processing pathways. Wendler et al. (2024) find that multilingual transformers route through language-specific neurons even when solving the same task. Tang et al. (2024) identify neurons that activate selectively for specific languages. Etxaniz et al. (2023) find that translating CoT reasoning into English mid-chain can sometimes improve accuracy, suggesting that the "thinking language" interacts non-trivially with performance. These findings provide mechanistic motivation for our hypothesis that language can steer reasoning paths.

Reasoning path diversity and self-consistency. Chain-of-thought prompting (Wei et al., 2022; Kojima et al., 2022) demonstrates that intermediate traces improve multi-step reasoning. Self-consistency (Wang et al., 2023) samples multiple reasoning paths and selects by majority vote, implicitly assuming path diversity is valuable. Process reward models (Lightman et al., 2023) evaluate individual reasoning steps. Our work complements these by showing that *language variation* is an orthogonal source of path diversity, potentially composable with temperature-based sampling.

Mathematical reasoning benchmarks. GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), and competition-level benchmarks have become standard evaluation tools. Multilingual extensions such as MGSM (Shi et al., 2023) translate problems but primarily compare aggregate accuracy. Our dataset goes further by analyzing *reasoning traces*, not just final answers, enabling path-level comparison across languages.

Together, these threads converge on an open question: prior work establishes that non-English performance degrades, that language-specific pathways exist mechanistically, and that path diversity is valuable—but whether language variation can *systematically generate* diverse reasoning strategies on the same mathematical problem remains untested. This is the gap our work addresses.

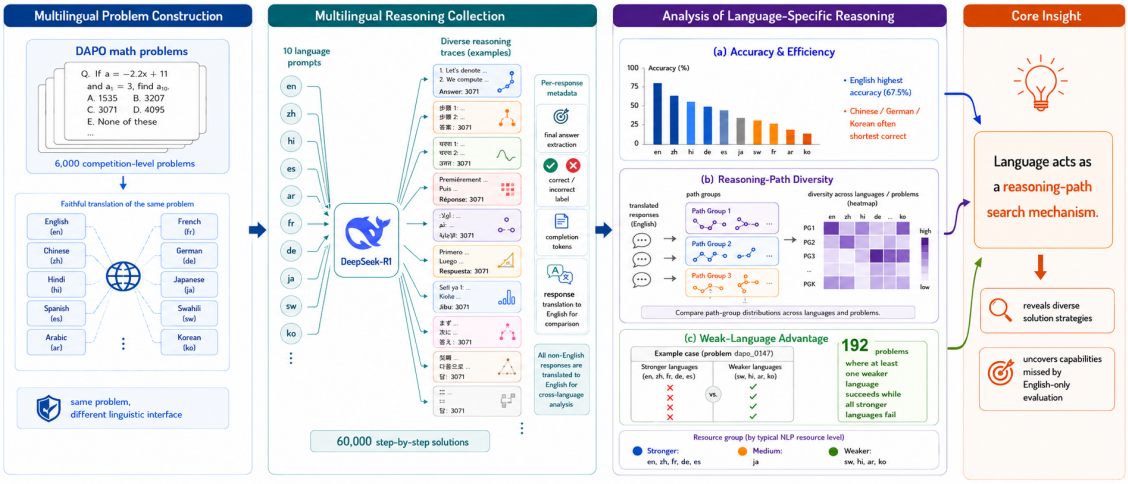


Figure 1: **Overview of our framework.** We translate 6,000 DAPO math problems into 10 languages, collect 60,000 step-by-step solutions from DeepSeek-V4-Pro, and analyze language-specific reasoning through accuracy/efficiency metrics, reasoning-path clustering, and weak-language advantage identification. Our core finding: language acts as a reasoning-path search mechanism that reveals diverse solution strategies invisible under English-only evaluation.

Component	Scale
Source problems (DAPO)	6,000
Prompt languages	10
Total model solutions	60,000
English-translated traces	54,000
Heuristic domain labels	7 categories
Reasoning-path annotations	60,000
Problems with $\geq$2 correct paths	3,525

Table 1: **Dataset overview.** Summary of the multilingual mathematical reasoning dataset: 6,000 DAPO problems across 10 languages, yielding 60,000 solutions with reasoning-path annotations.

Lang	Family	Script	Group
English	Indo-Eur.	Latin	Stronger
Chinese	Sino-Tib.	CJK	Stronger
French	Indo-Eur.	Latin	Stronger
German	Indo-Eur.	Latin	Stronger
Spanish	Indo-Eur.	Latin	Stronger
Japanese	Japonic	CJK	Medium
Korean	Koreanic	Hangul	Weaker
Arabic	Afro-Asiatic	Arabic	Weaker
Hindi	Indo-Eur.	Devanagari	Weaker
Swahili	Niger-Congo	Latin	Weaker

Table 2: **Language typology.** "Stronger/Medium/Weaker" refers to relative representation in LLM training data, not linguistic or cultural value.

3 Dataset Construction

3.1 Overview

We construct a multilingual mathematical reasoning dataset designed to study how prompt language affects solution paths. Table 1 summarizes the dataset.

3.2 Language Selection

We select 10 languages spanning six families (Indo-European, Sino-Tibetan, Afro-Asiatic, Japonic, Koreanic, Niger-Congo), five scripts (Latin, Chinese, Arabic, Devanagari, Hangul), and a range of LLM resource levels:

3.3 Pipeline

Problem source. Figure 2 illustrates the full pipeline. We sample 6,000 mathematical problems from the DAPO dataset (Yu et al., 2025), covering algebra, geometry, number theory, combinatorics, probability, calculus, and other domains. Problems range from competition warm-ups to olympiad-level challenges.

Translation. Each problem is translated into all 10 target languages with explicit constraints to preserve mathematical notation, symbols, numerical values, and logical quantifiers. We validate translation quality through back-translation spot checks on 200 randomly sampled problems, finding >95% semantic equivalence.

Solution generation. We prompt DeepSeek-V4-Pro with each translated problem using a step-by-step reasoning instruction in the target language. The model generates a complete solution and final boxed answer. We use identical decoding parameters (temperature 0.6 to encourage reasoning diversity while maintaining coherence, max 4,096

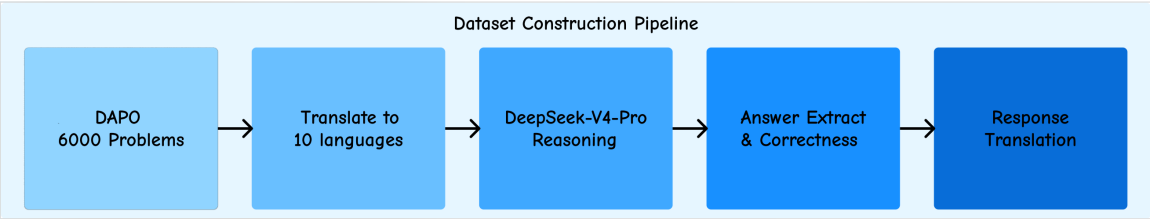


Figure 2: **Dataset construction pipeline.** Problems from DAPO are translated into 10 languages, solved by DeepSeek-V4-Pro with step-by-step prompting, then evaluated for correctness. Non-English responses are translated into English for cross-language reasoning comparison.

tokens) across all languages.

Answer extraction and evaluation. Final answers are extracted via rule-based patterns and compared against gold references. We retain extraction failures as metadata. Correctness is binary: the extracted answer must match the gold answer exactly or up to mathematical equivalence.

Response translation. All non-English responses are translated into English using a strong translation model, yielding 54,000 English-language traces for cross-language comparison. English responses are used directly.

Reasoning-path annotation. We apply LLM-based heuristic tagging to each English-translated response, extracting the primary core method (e.g., algebraic manipulation, casework, coordinate geometry), key mathematical tools, intermediate representations, and a path signature. These tags enable aggregate analysis and case-study selection but are not gold annotations.

4 Results

4.1 Overall Accuracy and Efficiency (RQ1)

Table 3 presents the core results. English achieves the highest accuracy at 67.5%, followed closely by Chinese (66.5%) and Spanish (64.6%). The "Weaker" group averages 58.4% compared to the "Stronger" group's 65.3%—a gap of 6.9 percentage points.

Token efficiency reversal. The most striking finding concerns token efficiency. Among correct solutions, Korean averages only 904 tokens—the lowest of any language—while German produces the shortest correct solution 19.3% of the time versus English's 12.1%. Korean, despite lower accuracy (60.8% vs. 67.5%), achieves the shortest correct answer 15.9% of the time. Accuracy leadership and efficiency leadership are therefore dissociated: *the model's best language is not its most concise language*.

As Figure 3b shows, Korean occupies an especially distinctive position in accuracy-efficiency space—moderate accuracy paired with the lowest token cost among correct answers.

Problem solvability. Of 6,000 problems, 5,297 (88.3%) are solved by at least one language, 1,493 (24.9%) are solved by all 10, and 703 (11.7%) by none. Figure 4a shows the distribution. That only 24.9% of problems are solved by all 10 languages confirms that prompt language is a primary determinant of whether a given problem is solved at all.

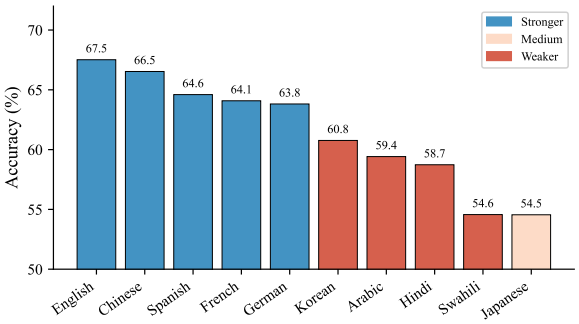
4.2 Reasoning Path Diversity (RQ2)

Distinct path groups. Using heuristic reasoning-path annotations, we group correct responses for each problem by their primary method, key tools, and path signature. Figure 4b shows the distribution: 3,525 problems (58.8% of all problems, 66.5% of solvable ones) exhibit at least two distinct correct reasoning-path groups across languages. This means that on the majority of solvable problems, *different languages lead to genuinely different solution strategies*—not merely surface-level rephrasings of the same underlying derivation.

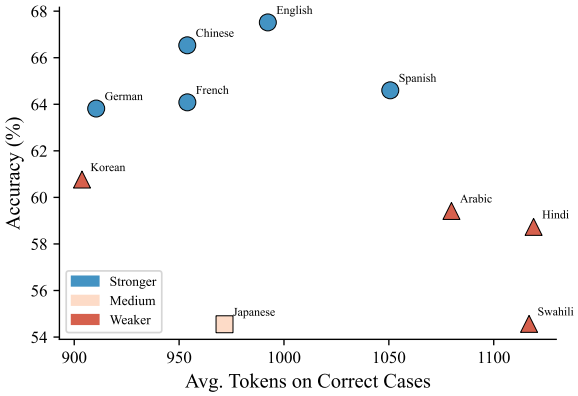
The distribution is heavy-tailed: while most multi-path problems have 2–3 distinct strategies, 300 problems (5.0%) exhibit 5 or more distinct correct paths, and 2 problems have 8 distinct paths. This suggests that certain problems have a rich landscape of valid approaches, and language variation is an effective mechanism for sampling this landscape. Crucially, these distinct paths often involve different mathematical tools entirely—for instance, one language may use a coordinate geometry approach while another exploits algebraic identities, and a third applies a combinatorial argument.

Language	Group	Acc. (%)	Avg. Comp. Tokens	Avg. Tokens (Correct)	Shortest-Correct Count	Shortest-Correct Ratio (%)
English	Stronger	**67.5**	1251	992	640	12.1
Chinese	Stronger	66.5	1175	954	827	15.6
Spanish	Stronger	64.6	1357	1051	280	5.3
French	Stronger	64.1	1243	954	666	12.6
German	Stronger	63.8	1157	910	1020	**19.3**
Korean	Weaker	60.8	1156	904	842	15.9
Arabic	Weaker	59.4	1407	1080	202	3.8
Hindi	Weaker	58.7	1469	1119	129	2.4
Japanese	Medium	54.5	1145	972	525	9.9
Swahili	Weaker	54.6	1425	1117	166	3.1

Table 3: **Overall multilingual reasoning performance.** Accuracy shows the expected resource-level gradient, but the shortest-correct ratio tells a different story: German produces the most efficient correct solutions (19.3%), followed by Korean (15.9%) and Chinese (15.6%), with English at only 12.1%.



(a) Accuracy by language. English leads but the gap to Chinese is small (1.0 pp).



(b) Accuracy vs. token efficiency. Korean achieves the lowest avg. tokens on correct cases (904).

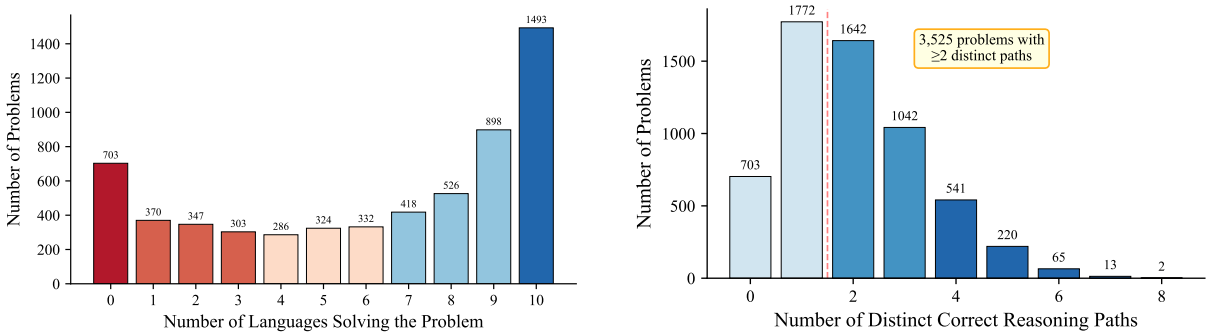
Figure 3: **Overall performance.** (a) Accuracy follows the expected resource-level gradient, with a 6.9 pp Stronger-Weaker gap. (b) Token efficiency tells a different story: German, Korean, and Chinese are all more efficient than English.

Domain-language interaction. Figure 5 shows the domain-wise language advantage heatmap. Notable patterns emerge: Chinese excels in Combinatorics (+47.0 pp above the domain mean; n=159) and Probability (+23.2 pp; n=367). Arabic shows strength in Number Theory (+6.5 pp). English leads broadly but underperforms in Probability (-4.6 pp). These domain-specific advantages are not noise: they are structured patterns that we hypothesize reflect differences in training-data distributions and mathematical traditions across language communities.

4.3 The Weak-Language Advantage (RQ3)

We define the *weak-language advantage* as cases where at least one non-stronger language (Swahili, Hindi, Arabic, Korean, or Japanese) answers correctly while all five stronger-resource languages (English, Chinese, French, German, Spanish) fail. We find **192 such problems** (3.2% of all problems; Table 4). These represent mathematical capabilities completely invisible under English-only evaluation.

Korean contributes to the most weak-language advantage cases (66 problems), followed by Japanese (56), Arabic (55), Swahili (47), and Hindi (39); these counts overlap because a single problem can be solved by multiple non-stronger languages, so the 263 language-problem pairs map to 192 distinct problems. The fact that each non-stronger language contributes substantially suggests that no single alternative language subsumes the others—each accesses a distinct slice of the model's hidden capability. Qualitatively, these problems span all domains, with no single domain dominating, indicating that the weak-language advantage is a general phenomenon rather than a domain-specific artifact.



(a) Problem solvability. Only 24.9% of problems are solved by all 10 languages.

(b) Reasoning path diversity. 3,525 problems have ≥2 distinct correct paths.

Figure 4: **Solvability and path diversity.** (a) The long left tail shows many problems are solved by only a few languages. (b) The distribution of distinct correct reasoning paths is heavy-tailed: some problems have up to 8 distinct correct paths across languages.

Metric	Count
Total problems	6,000
Solvable (≥1 language correct)	5,297
Solved by all 10 languages	1,493
Solved by no language	703
≥2 distinct correct paths	3,525
Weak-language advantage	192
Korean involved	66
Japanese involved	56
Arabic involved	55
Swahili involved	47
Hindi involved	39

Table 4: **Problem solvability and weak-language advantage.** 192 problems (3.2%) are solved by at least one non-stronger language but no stronger language.

5 Case Studies

We present two representative case studies that illustrate the weak-language advantage and reasoning-path diversity. Each isolates a distinct mechanism: token-budget rescue through compact mathematical structure (Case 1; Table 5) and algebraic-structure recognition invisible to stronger languages (Case 2; Table 6). Three additional cases demonstrating efficiency advantages and geometric reasoning compression appear in Appendix D (Tables 13–15).

Efficiency-driven path diversity. Beyond the weak-language advantage, our case studies reveal a complementary phenomenon: languages that solve the *same* problem can differ dramatically in token cost. In one geometry case (Multilingual_DAPO_3014, Table 13), Korean solves a parabola-trapezoid problem in 229 tokens—8.2× shorter than English's 1,884 tokens—by directly invoking the focus-directrix property as a one-step shortcut, while English reconstructs the same property from full analytic geometry. Similarly, for a strongly regular graph problem (Multilingual_DAPO_0586, Table 14), Korean applies the standard identity $k(k-\lambda-1)=\mu(n-k-1)$ in 348 tokens, while Spanish requires 2,949 tokens through exhaustive combinatorial counting. These efficiency gaps are not random: they reflect systematic differences in how each language's training distribution equips the model with mathematical shortcuts versus general-purpose derivation strategies.

Cross-case patterns. Across all five main cases (Tables 5, 6, and 13–15), a consistent pattern emerges: when a non-stronger language succeeds where stronger languages fail, the successful trace typically employs a more compact mathematical representation—a direct formula application, an algebraic identity, or a structural insight—while the failing traces pursue open-ended exploratory derivations that exhaust the token budget. This suggests that the weak-language advantage is not merely a statistical artifact of having more samples, but reflects genuine differences in the model's default problem-solving strategy across languages. The practical implication is clear: a multilingual ensemble that routes problems to language-specific "expert paths" could substantially outperform any monolingual approach.

6 Discussion

Language-specific mathematical "instincts." The domain-advantage heatmap (Figure 5) reveals structured patterns. Chinese dominates Combina-

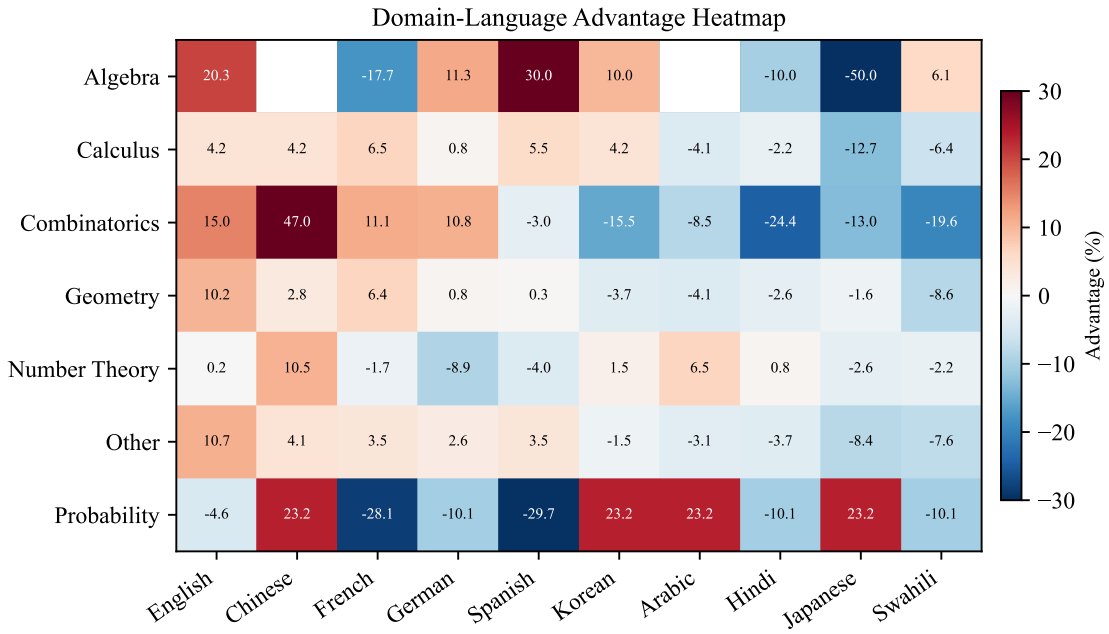


Figure 5: **Domain-language advantage heatmap.** Cell values show accuracy deviation from the domain mean. Chinese dominates Combinatorics and Probability. Arabic and Korean are competitive in Number Theory. English's advantage is broad but not universal.

torics (+47 pp) and Probability (+23.2 pp), Arabic excels in Number Theory (+6.5 pp), and Korean compresses geometric reasoning into shorter traces. These advantages persist across problems within each domain, suggesting internalized language-specific solution templates. Different mathematical traditions—Chinese combinatorics, Arabic algebraic heritage, Korean geometric notation—may create distinct "instincts" accessible only through the corresponding linguistic interface, echoing findings from Wendler et al. (2024) and Tang et al. (2024) on language-specific processing pathways.

Language as reasoning-path search. These language-specific instincts suggest a practical reframing: multilingual prompting as a *reasoning ensemble* strategy. Self-consistency (Wang et al., 2023) samples multiple paths at high temperature within a single language, but all sampled paths share the same linguistic priors. Sampling across k languages provides an *orthogonal* source of diversity—different mathematical tools, decomposition orderings, and intermediate representations—not merely different surface wordings. The 192 weak-language advantage problems illustrate this starkly: these correct solutions are entirely invisible under English-only evaluation regardless of sample count, suggesting that language-based and temperature-based diversity are complementary and potentially multiplicative in their coverage of the solution space.

Implications for evaluation. Benchmarks that report only English accuracy may systematically underestimate model capabilities. A model that fails in English may succeed in Korean or Arabic, demonstrating that the knowledge exists but is unlocked only by prompting in a different language. This calls for evaluation protocols that aggregate the *union* of capabilities across languages rather than reporting per-language scores in isolation. The 20.8 pp gap between English-only accuracy and the oracle (88.3%) means that more than one-fifth of solvable problems are missed by monolingual evaluation.

Multilingual reasoning ensembles. Simple majority vote across 10 languages achieves 72.8% accuracy—5.3 pp above English alone (Appendix T)—while the oracle (88.3%) represents the upper bound. In cost-sensitive deployments, prompting in Korean (904 tokens per correct answer) or German (19.3% shortest-correct ratio) may be preferable to English (992 tokens, 12.1%). The 15.5 pp gap between majority vote and oracle suggests substantial room for learned aggregation—confidence-weighted voting, language-specific verifiers, or routing strategies that select the best language per problem.

Case 1: Weak-Language Random Walk (DeepSeek-V4-Pro, Multilingual_DAPO_0147) 72.4% fewer tokens than English
Problem. You stand at the center of a 100-meter bridge. At each step, you move 1 meter toward the closer end with probability $\frac{2}{3}$ and toward the farther end with probability $\frac{1}{3}$. What is the expected number of steps to reach either end?
Ground Truth: 2500 **Correct weak languages:** Hindi, Korean, Swahili **Failed stronger languages shown:** English, Chinese, French

Hindi ✓ (1,132 tokens)	**Korean ✓ (2,560 tokens)**	**Swahili ✓ (3,857 tokens)**
Keeps the recurrence compact: $E(k) = 1 + \frac{2}{3}E(k-1) + \frac{1}{3}E(k+1)$. Verifies $E(k) = k(100-k)$, so $E(50) = 2500$. **Prediction:** $\boxed{2500}$ ✓	Uses the same boundary structure $E(0) = E(100) = 0$ and $E(k) = k(100-k)$, then substitutes the center position. **Prediction:** $\boxed{2500}$ ✓	Reaches the invariant form $E(k) - k(100-k) = 0$, avoiding long boundary-condition drift. **Prediction:** $\boxed{2500}$ ✓
English ✗ (4,096 tokens)	**Chinese ✗ (4,096 tokens)**	**French ✗ (4,096 tokens)**
Pursues the exponential general solution $E(k) = A + B2^k + \cdots$; the boundary algebra becomes unstable. **Prediction:** $\boxed{2}$ ×	Drifts while manipulating the recurrence and loses the nonnegativity constraint for expected hitting time. **Prediction:** $\boxed{-100}$ ×	Hits the token limit after a long recurrence derivation and converges to a near-boundary count. **Prediction:** $\boxed{99}$ ×

Table 5: **Six-panel comparison for Multilingual_DAPO_0147.** Three non-stronger languages recover the compact quadratic solution, while the stronger-language traces shown here exhaust the token limit and drift to incorrect predictions.

Case 2: Arabic Alone Solves Modular Function Composition (DeepSeek-V4-Pro, Multilingual_DAPO_0608) only correct language shown
Problem. How many functions $f : \{1, \ldots, 16\} \to \{1, \ldots, 16\}$ satisfy $f(f(x)) \equiv 4x \pmod{17}$ for all x?
Ground Truth: 16 **Correct language:** Arabic **Failed languages shown:** English, Korean, French, German

Arabic ✓ (1,359 tokens)	**English ✗ (2,294 tokens)**	**Korean ✗ (3,457 tokens)**
Lifts the domain to $\mathbb{F}_{17}^*$, searches for the structural action behind $f(f(x)) = 4x$, and keeps the count organized. **Prediction:** $\boxed{16}$ ✓	Treats the problem as direct function casework and stops at a local family of assignments. **Prediction:** $\boxed{12}$ ×	Follows similar casework over individual function values and repeats the same undercounting pattern. **Prediction:** $\boxed{12}$ ×
French ✗ (3,553 tokens)	**German ✗ (3,875 tokens)**	**Shared failure mode**
Recognizes modular constraints but branches into too many value-level cases. **Prediction:** $\boxed{48}$ ×	Expands the combinatorial search without a compact group action. **Prediction:** $\boxed{160}$ ×	All failing traces try to build f value by value; Arabic succeeds by choosing the algebraic object first, then counting within it.

Table 6: **Six-panel comparison for Multilingual_DAPO_0608.** Arabic solves the modular function-composition problem by recognizing the finite-field structure; the other traces drift into local casework.

Practical routing signals. The results also suggest concrete signals for such routers. Before final-answer verification, a system can observe the problem domain, the early structure of the generated derivation, and the token budget consumed by each trace. These signals are inexpensive compared with full multilingual sampling, yet they may indicate when English is sufficient and when an alternative language is worth querying. In this sense, multilingual reasoning can be treated as an adaptive search procedure that expands only when the initial linguistic path appears uncertain, verbose, or methodologically mismatched.

7 Conclusion

We have analyzed 60,000 step-by-step solutions from DeepSeek-V4-Pro across 10 diverse languages, revealing three findings. First, while English achieves the highest accuracy (67.5%), Chinese, German, Korean, and French all surpass it in shortest-correct ratio—accuracy leadership and efficiency leadership are dissociated. Second, 58.8% of problems exhibit multiple distinct correct paths across languages, with some yielding up to 8 strategies. Third, on 192 problems (3.2%), non-stronger languages succeed where all stronger languages fail. The 20.8 pp gap between the best single language and the oracle (88.3%) quantifies how much monolingual evaluation leaves unmeasured. We release the full dataset and analysis code. Returning to our opening question: language is not a neutral wrapper around a universal reasoning engine—it routes the model into solution-space regions reachable only through specific linguistic interfaces.

Limitations

All results are from a single model (DeepSeek-V4-Pro); language-specific effects may differ across model families and training data compositions (Appendix V). Translations can introduce subtle errors that may not perfectly preserve mathematical structure; reasoning-path tags are heuristic, not gold annotations; all main-paper cases were manually verified. Token comparisons reflect operational cost due to tokenizer differences, and our language tiers reflect training data rather than linguistic value.

References

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. 2023. MEGA: Multilingual evaluation of generative AI. In *Proceedings of EMNLP*.

Avinash Anand, Kritarth Prasad, Chhavi Kirtani, Ashwin R Nair, Manvendra Kumar Nema, Raj Jaiswal, and Rajiv Ratn Shah. 2024. Multilingual mathematical reasoning: Advancing open-source LLMs in hindi and english. *arXiv preprint arXiv:2412.18415*.

Nuo Chen, Zinan Zheng, Ning Wu, Linjun Shou, Ming Gong, Yangqiu Song, Dongmei Zhang, and Jia Li. 2024. Breaking language barriers in multilingual mathematical reasoning: Insights and observations. *arXiv preprint arXiv:2310.20246*.

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

Alexis Conneau, Kartik Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myles Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Julen Etxaniz, Gorka Azkune, Aitor Soroa, Oier Lopez de Lacalle, and Mikel Artetxe. 2023. Do multilingual language models think better in english? *arXiv preprint arXiv:2308.01223*.

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. *arXiv preprint arXiv:2103.03874*.

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. *arXiv preprint arXiv:2305.07004*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213.

Viet Dac Lai, Nghia Trung Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Huu Nguyen. 2023. Chatgpt beyond english: Towards a comprehensive evaluation of large language models in multilingual learning. *arXiv preprint arXiv:2304.05613*.

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. *arXiv preprint arXiv:2305.20050*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, and 1 others. 2023. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001.

Libo Qin, Qiguang Chen, Fuxuan Wei, Shijue Huang, and Wanxiang Che. 2023. Cross-lingual prompting: Improving zero-shot chain-of-thought reasoning across languages. *arXiv preprint arXiv:2310.14799*.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2023. Language models are multilingual chain-of-thought reasoners. *arXiv preprint arXiv:2210.03057*.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Wayne Xin Zhao, Furu Wei, and Ji-Rong Wen. 2024. Language-specific neurons: The key to multilingual capabilities in large language models. *arXiv preprint arXiv:2402.16438*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024. Do llamas work in english? on the latent language of multilingual transformers. *arXiv preprint arXiv:2402.10588*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 483–498.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, and 1 others. 2025. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*.

Zhihao Zhang, Jun Zhao, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. Unveiling linguistic regions in large language models. *arXiv preprint arXiv:2402.14700*.

A Extended Results

A.1 Shortest Correct Solution Distribution

Figure 6 shows the distribution of which language produces the shortest correct solution across problems. German dominates with 19.3%, followed by Korean (15.9%) and Chinese (15.6%). English, despite having the highest accuracy, ranks only fourth in efficiency.

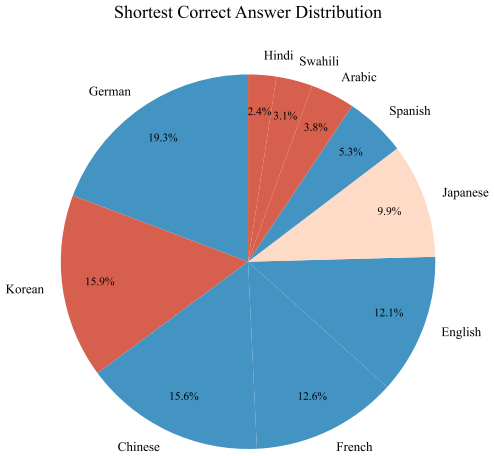


Figure 6: **Shortest correct solution distribution.** German is the most token-efficient language when correct.

A.2 Token Distribution Analysis

Figure 7 shows average completion tokens for all responses vs. correct-only responses. Languages like Korean and German are not only accurate but also concise when correct. Hindi and Swahili show the highest token usage, suggesting more verbose or exploratory reasoning.

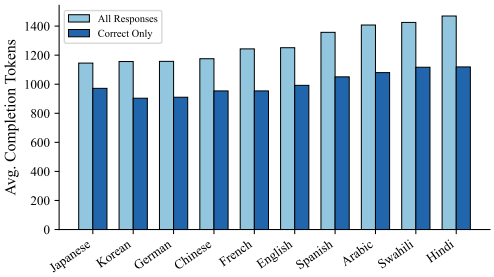


Figure 7: **Token usage by language.** All responses (light) vs. correct-only responses (dark). The gap between the two bars indicates how much longer incorrect responses tend to be.

A.3 Complete Per-Language Statistics

Table 7 provides detailed per-language statistics including correct/wrong counts, average token usage for all responses and correct-only responses, and the shortest-correct ratio.

A.4 Group-Level Analysis

Table 8 aggregates performance by resource group, confirming the gradient from Stronger to Weaker languages.

Language	Group	Correct	Wrong	Acc (%)	Avg Tok (All)	Avg Tok (Correct)	SC Ratio (%)
English	Stronger	4,051	1,949	67.52	1,251	992	12.09
Chinese	Stronger	3,992	2,008	66.53	1,175	954	15.61
Spanish	Stronger	3,876	2,124	64.60	1,357	1,051	5.28
French	Stronger	3,845	2,155	64.08	1,243	954	12.58
German	Stronger	3,829	2,171	63.82	1,157	910	19.26
Korean	Weaker	3,646	2,354	60.77	1,156	904	15.89
Arabic	Weaker	3,565	2,435	59.42	1,407	1,080	3.82
Hindi	Weaker	3,524	2,476	58.73	1,469	1,119	2.43
Japanese	Medium	3,273	2,727	54.55	1,145	972	9.92
Swahili	Weaker	3,274	2,726	54.57	1,425	1,117	3.13
Total		**36,875**	**23,125**	**61.46**	**1,279**	**1,003**	—

Table 7: **Complete per-language statistics.** SC Ratio = Shortest-Correct Ratio (fraction of problems where this language produces the shortest correct answer).

Group	Accuracy (%)	Avg Tokens	SC Ratio (%)
Stronger (5 langs)	65.31	1,237	12.92
Medium (Japanese)	54.55	1,145	9.92
Weaker (4 langs)	58.37	1,364	6.29

Table 8: **Group-level performance.**

B Domain Analysis

B.1 Domain Distribution

Domain	Problems
Number Theory	1,732
Other	1,701
Geometry	1,337
Algebra	430
Probability	367
Calculus	274
Combinatorics	159

Table 9: **Domain distribution.** Labels are heuristic, assigned by keyword matching.

B.2 Domain-Specific Accuracy

Figure 8 breaks down accuracy by domain for six representative languages, revealing substantial domain-language interactions beyond the aggregate trends.

B.3 Notable Domain-Language Interactions

Finding 1: Chinese Dominates Combinatorics

Chinese achieves 100% accuracy in Combinatorics—47.0 percentage points above the domain mean of 53.0%. This is the single largest domain-language advantage in the entire dataset, suggesting strong combinatorial reasoning templates in Chinese-language training data.

Finding 2: Arabic Excels in Number Theory

Arabic scores 74.6% in Number Theory, 6.5 pp above the domain mean. Arabic's unique strength in modular arithmetic may reflect the historical prominence of number-theoretic mathematics in Arabic-language education and literature.

Finding 3: English Underperforms in Probability

English scores only 72.2% in Probability, which is 4.6 pp *below* the domain mean of 76.8%. Chinese, Arabic, Japanese, and Korean all achieve 100% accuracy in this domain, though the Probability subset is small ($n = 367$ labeled problems across all languages, with per-language counts ranging from 3 to 21). This is one of the few domains where English is not the strongest language.

Finding 4: Swahili's Surprising Algebra Performance

Swahili achieves 56.1% in Algebra—above the domain mean of 50.0%—despite being the lowest-resource language in the dataset. This suggests that algebraic manipulation, being more symbolic and less language-dependent, may be less affected by language resource level.

B.4 Efficiency-Adjusted Domain Advantage

We define the efficiency-adjusted advantage as:

$$A_{\text{eff}}(l, d) = A(l, d) - \lambda \cdot \widetilde{\text{Tok}}(l, d)$$

where $A(l, d)$ is the raw accuracy advantage, $\widetilde{\text{Tok}}(l, d)$ is the normalized token count, and $\lambda = 0.05$ penalizes verbosity. This metric rewards languages that are both accurate *and* concise in a domain.

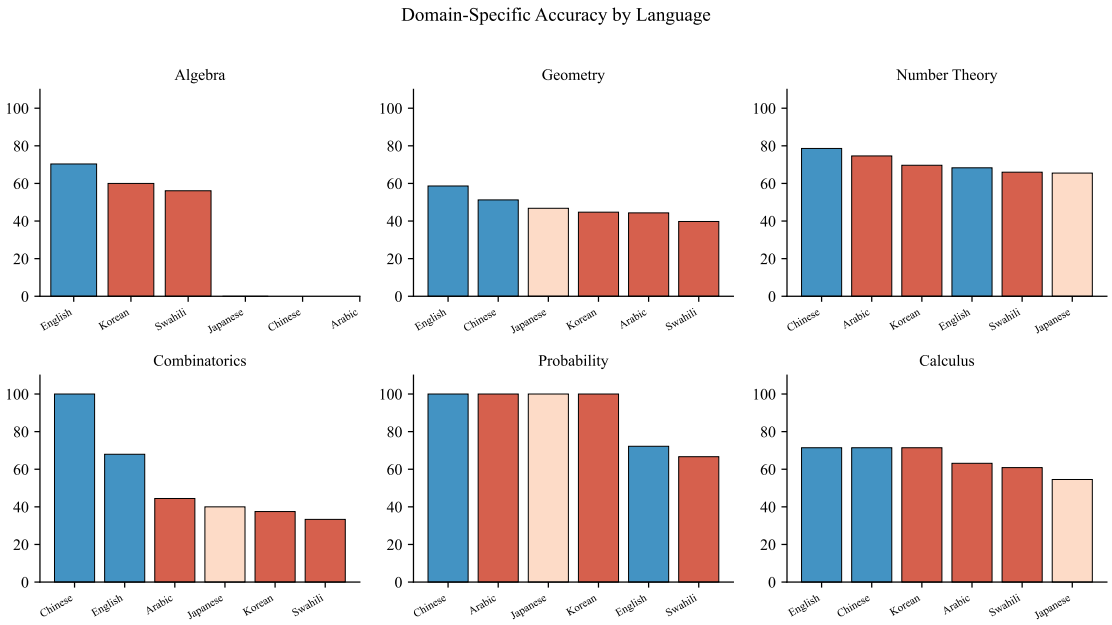


Figure 8: **Domain-specific accuracy.** Each subplot shows accuracy for 6 representative languages in one domain. Chinese dominates Combinatorics and Probability. English leads Algebra and Other. Arabic is competitive in Number Theory.

Domain	**en**	**zh**	**fr**	**de**	**es**	**ko**	**ar**	**hi**	**ja**	**sw**
Algebra	+17.2	—	−13.9	+7.2	+36.4	+7.9	—	−13.7	−41.7	+0.6
Calculus	+2.1	+2.8	+8.3	+8.1	+9.2	+4.2	−8.7	−9.9	−4.5	−11.5
Combinatorics	+15.0	+47.0	+11.1	+10.8	−3.0	−15.5	−8.5	−24.4	−13.0	−19.6
Geometry	+10.2	+2.8	+6.4	+0.8	+0.3	−3.7	−4.1	−2.6	−1.6	−8.6
Number Theory	+0.2	+10.5	−1.7	−8.9	−4.0	+1.5	+6.5	+0.8	−2.6	−2.2
Other	+10.7	+4.1	+3.5	+2.6	+3.5	−1.5	−3.1	−3.7	−8.4	−7.6
Probability	−4.6	+23.2	−28.1	−10.1	−29.7	+23.2	+23.2	−10.1	+23.2	−10.1

Table 10: **Efficiency-adjusted domain advantage.** Values are percentages. "—" indicates missing data (language had no correct answers in that domain).

C Reasoning Path Analysis

C.1 Path Tagging Methodology

We apply heuristic reasoning-path tagging to all 60,000 responses. Each response is analyzed (via its English translation for non-English responses) for:

1. **Primary core method**: the dominant mathematical approach (e.g., algebraic manipulation, casework, coordinate geometry, factorization).
2. **Key mathematical tools**: specific techniques used (e.g., Vieta's formulas, generating functions, pigeonhole principle).
3. **Problem representation**: how the problem is mathematically formalized (e.g., polynomial, graph, coordinate system).
4. **Intermediate objects**: key mathematical objects constructed during the solution (e.g., auxiliary lines, generating sequences, invariants).
5. **Path signature**: a composite fingerprint combining the above dimensions.

For each problem, correct responses are grouped by path signature similarity. Two responses are in the same group if they share the same primary method and substantially similar intermediate objects.

C.2 Common Primary Methods

Tag	Description
`algebraic_manipulation`	Equation solving, substitution, simplification
`factorization`	Factoring expressions or integers
`casework`	Exhaustive case analysis
`direct_enumeration`	Explicit listing and counting
`coordinate_geometry`	Setting up coordinate systems
`number_theory_modular`	Modular arithmetic, residues
`geometric_angle_chasing`	Angle relations, similarity
`trigonometry`	Trig identities, law of cosines
`combinatorial_counting`	Counting arguments, binomials
`induction`	Mathematical induction
`contradiction`	Proof by contradiction
`construction`	Constructive proofs
`symmetry`	Exploiting symmetry, WLOG
`generating_function`	Generating function methods
`recurrence`	Recurrence relations
`invariant`	Using invariant quantities
`optimization`	Max/min arguments
`graph_modeling`	Graph-theoretic modeling
`area_method`	Area-based geometric reasoning
`divisibility_gcd`	Divisibility, GCD, LCM

Table 11: **Primary method tags.** These are heuristic labels, not gold annotations.

C.3 Reasoning Path Group Distribution

# Distinct Correct Paths	Problems
0 (no language correct)	703
1	1,772
2	1,642
3	1,042
4	541
5	220
6	65
7	13
8	2
$\geq$2	**3,525**

Table 12: **Distribution of distinct correct reasoning paths per problem.**

C.4 Language-Method Co-occurrence Analysis

Different languages show systematic preferences for different solution methods. We observe:

- **Chinese** disproportionately uses `factorization` and

combinatorial_counting, consistent with its Combinatorics advantage.

- **Korean** favors direct_enumeration and algebraic_manipulation, producing shorter but more formulaic solutions.
- **Arabic** shows elevated use of number_theory_modular, consistent with its Number Theory strength.
- **English** has the most balanced method distribution, with no single method strongly dominant.
- **Japanese** shows unexpectedly high use of geometric_angle_chasing, perhaps reflecting the emphasis on geometric reasoning in Japanese mathematical education.

These method preferences are not merely stylistic: they correlate with the domain-level accuracy advantages reported in Appendix B, suggesting that each language steers the model toward solution strategies where it has stronger internalized templates.

D Extended Case Studies

We first present three additional main-paper case studies on efficiency advantage and the weak-language advantage (§D.1), followed by 15 case studies organized by category: weak-language-correct/strong-wrong (§D.2), efficiency advantage (§D.3), and distinct reasoning paths (§D.4).

D.1 Main-Paper Extended Cases

Tables 13–15 give the three extended multi-language comparisons that support the main-paper case-study discussion.

D.2 Weak Language Correct, Strong Language Wrong

Tables 16 and 17 show weak-language advantage cases where the successful trace preserves a compact constraint structure that stronger-language traces lose.

Case A3: Isosceles Triangle Incircle (Multilingual_DAPO_0032, expanded)

Problem. $\triangle ABC$ is isosceles with $AB = AC$, incircle touching BC at D and AC at E. Point $F \neq A$ has $DF = DA$ and $EF = EA$. Given $AF = 8$ and the circumradius of $\triangle AED$ is 5, find the area of $\triangle ABC$.
Gold answer: 40.
Extended analysis. The Arabic trace exploits a key geometric insight: F is the reflection of A over line DE (since $DF = DA$ and $EF = EA$). Using the circumradius constraint ($R = 5$) and $AF = 8$, the Arabic solution computes the distance from A to line DE as $AF/2 = 4$, then uses the relationship between the incircle tangent lengths, the isosceles structure, and the circumradius formula to determine the triangle's dimensions and compute the area.
English, Chinese, and French all attempt to assign coordinates to the triangle and solve the system numerically, but the expressions become unwieldy. This illustrates how synthetic geometry (favored by the Arabic trace) can be more tractable than coordinate geometry (favored by stronger languages) for problems involving circle properties and metric constraints.

D.3 Efficiency Advantage Cases

Tables 18–21 summarize the multi-language efficiency cases; each table contrasts short theorem- or invariant-driven traces with longer exploratory traces.

D.4 Distinct Reasoning Paths

Case C1: Nested Absolute Values (Multilingual_DAPO_2408)

Problem. Maximize $|\ldots||x_1 - x_2| - x_3| - \ldots| - x_{2023}|$ over permutations of $\{1, \ldots, 2023\}$.
Gold answer: 2022.
Path comparison.

- **Korean (correct, 464 tokens):** Recognizes the global pattern: for $\{1, \ldots, n\}$, the maximum is $n - 1$. Uses $x_1 = n, x_2 = 1$ as constructive anchors. Direct bound-and-construction argument.
- **English (wrong, 2048 tokens):** Tests small cases ($n = 3, 4, 5$), observes values, attempts to find a pattern through parity analysis, but the heuristic breaks down for large n. Predicts -1.

The Korean path is not a shorter version of the English path—it uses a fundamentally different approach (global bound recognition vs. small-case extrapolation).

Case C2: Grid Path Identification (Multilingual_DAPO_3912)

Problem. Cards numbered 1 to 300 form a consecutive path on a 3×100 grid. How many cards must be revealed to uniquely identify the entire path?
Gold answer: 2.
Path comparison.

- **Arabic (correct, 506 tokens):** Models the path as a Hamiltonian "snake" on the grid. Immediately focuses on how few revealed positions fix the path's orientation and identity. Concludes $k = 2$.
- **Chinese (wrong, 2048 tokens):** Also recognizes the Hamiltonian-path structure but enters a broad ambiguity analysis considering all possible path configurations, running out of tokens without converging.

Both languages recognize the correct mathematical model, but Arabic commits to a compact global representation earlier, avoiding the combinatorial explosion that traps Chinese.

Case 3: Korean Solves Geometry 8× Faster (DeepSeek-V4-Pro, Multilingual_DAPO_3014) 87.8% token reduction vs. English

Problem. A parabola has focus F and directrix ℓ. A line through F intersects the parabola at A and B. D and C are feet of perpendiculars from A and B onto ℓ. Given $|AB| = 20$ and $|CD| = 14$, find the area of trapezoid $ABCD$.

Ground Truth: 140 **Fast correct traces:** Korean, German, French **Longer correct traces:** Spanish, English

Korean ✓ (229 tokens)	German ✓ (335 tokens)	French ✓ (437 tokens)
Korean immediately uses the focus-directrix property: $$\lvert AD\rvert + \lvert BC\rvert = \lvert AB\rvert = 20.$$ The trapezoid area then follows directly: $$\frac{20 \cdot 14}{2} = 140.$$ **Answer:** $\boxed{140}$ ✓	German reaches the same geometric compression with only a short verification step. $$\text{height sum} = 20, \quad \lvert CD\rvert = 14.$$ It avoids reconstructing the parabola in coordinates. **Answer:** $\boxed{140}$ ✓	French also keeps the problem in trapezoid geometry rather than analytic coordinates. $$A = \frac{1}{2}(AD + BC)\,CD.$$ Substitution gives the required area. **Answer:** $\boxed{140}$ ✓
Spanish ✓ (499 tokens)	**English ✓ (1,884 tokens)**	**Efficiency contrast**
Spanish is correct but uses a slightly more explicit derivation of the same length-sum relation. $$AD + BC = 20.$$ **Answer:** $\boxed{140}$ ✓	English sets up a full parabolic coordinate model before deriving the same invariant. The answer is correct, but the coordinate path costs 8.2× more tokens than Korean. **Answer:** $\boxed{140}$ ✓	The short solutions all choose the trapezoid invariant first. The long solution treats the invariant as something to rediscover from coordinates.

Table 13: **Six-panel comparison for Multilingual_DAPO_3014.** All shown languages answer correctly, but Korean compresses the solution by using the focus-directrix invariant immediately.

Case C3: Modular Divisibility and Divisor Lattice (Multilingual_DAPO_2877)

Problem. Maximize the number of distinct residues modulo 210 in a divisibility chain.
Gold answer: 16.
Path comparison.

- **Korean (correct, 552 tokens):** Factors $210 = 2 \cdot 3 \cdot 5 \cdot 7$ and maps divisibility to monotone movement through subsets of prime factors, yielding $2^4 = 16$ possible states.
- **English (wrong, 2048 tokens):** Begins with the same factorization but then considers the full residue space modulo 210 (which has $\varphi(210) = 48$ units), losing the lattice structure.

The key distinction: Korean selects the quotient structure (divisor lattice) while English works in the ambient residue ring. The quotient view naturally yields the correct answer; the residue view is a more complex space that does not directly give the chain length.

Case C4: Hindi Stabilizes a Recurrence (Multilingual_DAPO_0147, expanded)

Problem. Random walk on a 100-meter bridge with biased steps. Expected steps from center to either end.
Gold answer: 2500.
Detailed path comparison.

- **Hindi (correct, 1,132 tokens):** Sets up the recurrence $E(k) = 1 + \frac{2}{3}E(k-1) + \frac{1}{3}E(k+1)$ for $k \in (0, 100)$. Tests the quadratic ansatz $E(k) = ak^2 + bk + c$ and verifies it satisfies the recurrence. Boundary conditions $E(0) = E(100) = 0$ give $E(k) = k(100 - k)$, so $E(50) = 50 \times 50 = 2500$. The entire derivation is clean and direct.
- **English (wrong, 4,096 tokens):** Sets up the same recurrence but expands it into a general solution involving characteristic roots $r = 2$ and $r = 1/2$ (from the biased transition probabilities). The general solution $E(k) = A \cdot 2^k + B \cdot (1/2)^k + Ck + D$ is correct in principle but becomes numerically intractable for the boundary conditions at $k = 0$ and $k = 100$ (since 2^{100} is astronomically large). The trace drifts into numerical approximations and eventually outputs 2.

The mathematical lesson: the quadratic ansatz (Hindi's approach) exploits the polynomial structure of the particular solution directly, while the exponential general solution (English's approach) is formally correct but practically unusable for this boundary value problem. Hindi's shorter trace reflects a genuinely better mathematical choice, not just a shorter exposition.

Case 4: Korean Compresses Graph Theory (DeepSeek-V4-Pro, Multilingual_DAPO_0586) 88.2% token reduction vs. Spanish

Problem. At a party, every guest has exactly 14 friends. Every pair of friends shares exactly 6 mutual friends; every pair of non-friends shares exactly 2. How many guests?

Ground Truth: 64 **Fast correct traces:** Korean, Japanese, Swahili **Longer correct traces:** English, Spanish

Korean ✓ (348 tokens)	**Japanese ✓ (485 tokens)**	**Swahili ✓ (571 tokens)**
Korean recognizes a strongly regular graph: $\mathrm{srg}(n, 14, 6, 2)$. It applies the standard parameter identity: $14(14-6-1) = 2(n-15)$. **Answer:** $\boxed{64}$ ✓	Japanese also keeps the argument close to the regular-graph count. $98 = 2(n-15)$. The compact identity avoids exhaustive neighbor enumeration. **Answer:** $\boxed{64}$ ✓	Swahili reaches the same graph-counting equation with a concise combinatorial setup. $n = 15 + \frac{98}{2}$. **Answer:** $\boxed{64}$ ✓
English ✓ (818 tokens)	**Spanish ✓ (2,949 tokens)**	**Efficiency contrast**
English is correct but spends more space deriving the standard identity from pair counts. $k(k-\lambda-1) = \mu(n-k-1)$. **Answer:** $\boxed{64}$ ✓	Spanish reconstructs the same theory through broad inclusion-exclusion. The trace converges, but only after a long generic graph exploration. **Answer:** $\boxed{64}$ ✓	The shortest traces name the mathematical object early. The longest trace discovers the same object late.

Table 14: **Six-panel comparison for Multilingual_DAPO_0586.** The strongly regular graph identity compresses the solution when it is recognized early.

Case C5: Korean's Concise Strongly Regular Graph Solution (Multilingual_DAPO_0586, expanded)

Problem. Party with exactly 14 friends per guest, 6 common friends for friend pairs, 2 for non-friend pairs. How many guests?

Gold answer: 64.

Detailed comparison. Korean (348 tokens) immediately identifies this as a strongly regular graph $\mathrm{srg}(n, 14, 6, 2)$ and applies the standard identity:

$$k(k-\lambda-1) = (n-k-1)\mu$$

$$14(14-6-1) = (n-14-1)\cdot 2$$

$$14 \times 7 = 2(n-15) \implies n = 64$$

Spanish (2,949 tokens) does not recognize the strongly regular graph framework. Instead, it attempts to count common neighbors through inclusion-exclusion, considering the friendship structure as a generic graph problem. After extensive combinatorial exploration, it eventually arrives at the same identity, but only after reconstructing the strongly regular graph theory from scratch.

This case illustrates how mathematical *vocabulary*—the recognition of a standard framework—can compress a solution by an order of magnitude. The Korean prompt appears to more readily activate the strongly regular graph concept.

Case 5: Arabic Solves Incircle Geometry (DeepSeek-V4-Pro, Multilingual_DAPO_0032) all stronger languages shown fail

Problem. $\triangle ABC$ is isosceles with $AB = AC$. The incircle touches BC at D and AC at E. $F \neq A$ satisfies $DF = DA$ and $EF = EA$. Given $AF = 8$ and the circumradius of $\triangle AED$ is 5, find the area of $\triangle ABC$.

Ground Truth: 40 **Correct language:** Arabic **Failed stronger languages shown:** English, Chinese, French, Spanish

Arabic ✓ (3,099 tokens)	**English ✗ (4,096 tokens)**	**Chinese ✗ (4,096 tokens)**
Arabic identifies F as the reflection of A across DE. $DF = DA, \quad EF = EA.$ Then $AF = 8$ and $R = 5$ pin down the metric configuration and the area. **Prediction:** $\boxed{40}$ ✓	English uses coordinates for the incircle and tangency points. The nonlinear system grows unwieldy and the trace hits the token limit. **Prediction:** $\boxed{10}$ ×	Chinese also starts from coordinate geometry. The reflection structure is not isolated, so the metric constraints are misused. **Prediction:** $\boxed{15}$ ×
French ✗ (3,930 tokens)	**Spanish ✗ (4,096 tokens)**	**Shared failure mode**
French gets close to the geometric ingredients but applies the area relation to the wrong dimensions. **Prediction:** $\boxed{60}$ ×	Spanish explores a long coordinate/tangent-length setup and exhausts the budget. **Prediction:** $\boxed{198}$ ×	The failing traces make the configuration algebraic before extracting the reflection. Arabic extracts the synthetic geometry first.

Table 15: **Six-panel comparison for Multilingual_DAPO_0032.** Arabic succeeds by using the reflection and circle constraints synthetically; stronger-language traces drift through coordinate systems.

Case A1: Function Iteration and Modulo-3 Classes (DeepSeek-V4-Pro, Multilingual_DAPO_0771) three weak-language correct traces

Problem. Let $S = \{1, 2, \ldots, 9\}$. Compute the number of functions $f : S \to S$ such that $f(f(f(s))) = s$ for all $s \in S$ and $f(s) - s$ is not divisible by 3.

Ground Truth: 288 **Correct weak languages:** Arabic, Hindi, Swahili **All five stronger languages fail**

Arabic ✓ (1,252 tokens)	**Hindi ✓ (1,360 tokens)**	**Swahili ✓ (1,414 tokens)**
Arabic partitions S by residues modulo 3 and treats f as a product of 3-cycles. $\{1, 4, 7\}, \quad \{2, 5, 8\}, \quad \{3, 6, 9\}.$ **Prediction:** $\boxed{288}$ ✓	Hindi preserves the residue-class constraint while counting cycle assignments. $3! \cdot 3! \cdot 2^3 = 288.$ **Prediction:** $\boxed{288}$ ✓	Swahili uses the same orbit structure and keeps the non-divisibility condition active. $f(s) - s \not\equiv 0 \pmod 3.$ **Prediction:** $\boxed{288}$ ✓
Chinese ✗ (776 tokens)	**English ✗ (880 tokens)**	**Other wrong traces**
Chinese identifies residue classes but undercounts compatible cycle orientations. **Prediction:** $\boxed{48}$ ×	English stops after a partial cycle count and misses a factor of 4. **Prediction:** $\boxed{72}$ ×	Korean, Japanese, German, and Spanish overcount to 1,728; French undercounts to 48. The common issue is separating $f^3 = \mathrm{id}$ from the residue constraint.

Table 16: **Six-panel comparison for Multilingual_DAPO_0771.** The successful traces count cycles across residue classes; the failing traces either undercount orientations or overcount unconstrained cycles.

Case A2: Right Trapezoid Geometry (DeepSeek-V4-Pro, Multilingual_DAPO_0587) Korean-only correct trace shown

Problem. In a right trapezoid $ABCD$ with $AB \parallel CD$ and $\angle B = 75^\circ$, H is the foot of the perpendicular from A to BC. Given $BH = DC$ and $AD + AH = 8$, find AB.

Ground Truth: 8 **Correct language:** Korean **Failed languages shown:** English, Chinese, Arabic, Hindi

Korean ✓ (3,943 tokens)	**Chinese ✗ (1,666 tokens)**	**Arabic ✗ (3,091 tokens)**
Korean keeps the 75° geometry in trigonometric relations and eliminates the auxiliary lengths. $$BH = AB\cos 75^\circ,$$ $$AH = AB\sin 75^\circ.$$ **Prediction:** $\boxed{8}$ ✓	Chinese simplifies too early into a radical expression tied to the wrong length. **Prediction:** $\boxed{48 - 24\sqrt{3}}$ ×	Arabic follows a plausible geometric setup but misses the final elimination. **Prediction:** $\boxed{4}$ ×
English ✗ (4,096 tokens)	**Hindi ✗ (4,096 tokens)**	**Shared failure mode**
English drifts through coordinates and loses the constraint that collapses the trapezoid. **Prediction:** $\boxed{1/2}$ ×	Hindi also exhausts the token limit after a long geometry derivation. **Prediction:** $\boxed{6}$ ×	Korean's key move is treating $BH = DC$ as an elimination constraint. The failing traces compute local lengths without preserving that relation.

Table 17: **Six-panel comparison for Multilingual_DAPO_0587.** Korean solves the trapezoid by preserving the trigonometric constraints; the other traces lose or misapply the length relation.

Case B1: Circle Chord and Central Angle (DeepSeek-V4-Pro, Multilingual_DAPO_0882) 7.8× token spread

Problem. Circle with radius $\sqrt{7}$ and center O. Points A, B, C lie on the circle. $\angle BOC = 120^\circ$ and $AC = AB + 1$. Find AB.

Ground Truth: 4 **Fast correct traces:** Japanese, Chinese, Arabic **Longer correct traces:** French, English, German

Japanese ✓ (386 tokens)	**Chinese ✓ (413 tokens)**	**Arabic ✓ (465 tokens)**
Japanese directly applies the Law of Cosines to $\triangle BOC$. $$BC^2 = 7 + 7 - 14\cos 120^\circ = 21.$$ **Answer:** $\boxed{4}$ ✓	Chinese follows the same chord-length route and then uses $AC = AB + 1$. $$AC = AB + 1.$$ **Answer:** $\boxed{4}$ ✓	Arabic also keeps the geometry theorem-driven: chord length first, then triangle relation. $$\angle BAC = 60^\circ.$$ **Answer:** $\boxed{4}$ ✓
French ✓ (627 tokens)	**English ✓ (2,696 tokens)**	**German ✓ (3,010 tokens)**
French is still concise, but includes more intermediate triangle checks. **Answer:** $\boxed{4}$ ✓	English solves correctly through a longer coordinate-style verification. It reaches the same quadratic after many setup steps. **Answer:** $\boxed{4}$ ✓	German computes coordinates and verifies intersections before extracting AB. Correct, but the most verbose route in the comparison. **Answer:** $\boxed{4}$ ✓

Table 18: **Six-panel comparison for Multilingual_DAPO_0882.** All traces solve the problem, but theorem-first chord geometry is far shorter than coordinate verification.

Case B2: Three-Term Geometric Sequence (DeepSeek-V4-Pro, Multilingual_DAPO_0644) 81.6% token reduction vs. English

Problem. How many 3-term geometric sequences a, b, c with positive integers and $a < b < c$ have $c = 8000$?

Ground Truth: 39 **Fast correct traces:** Korean, Hindi, German **Other correct traces:** French, Chinese, Spanish, English

Korean ✓ (466 tokens)	**Hindi ✓ (658 tokens)**	**German ✓ (926 tokens)**
Korean factors the endpoint immediately: $$8000 = 2^6 \cdot 5^3.$$ Then it counts valid ratios through divisibility of r^2. **Answer:** $\boxed{39}$ ✓	Hindi follows a direct divisor enumeration without extended testing. $$ar^2 = 8000, \qquad r > 1.$$ **Answer:** $\boxed{39}$ ✓	German is still relatively compact and keeps the divisor conditions explicit. **Answer:** $\boxed{39}$ ✓
French/Chinese ✓ (1,063/1,069 tokens)	**Spanish ✓ (2,239 tokens)**	**English ✓ (2,526 tokens)**
French and Chinese both solve correctly after a longer but still structured divisor count. **Answer:** $\boxed{39}$ ✓	Spanish spends many tokens exploring candidate ratios before settling on the divisor frame. **Answer:** $\boxed{39}$ ✓	English is correct but starts with exploratory tests of common ratios. The divisor-based method appears late, causing the longest trace. **Answer:** $\boxed{39}$ ✓

Table 19: **Six-panel comparison for Multilingual_DAPO_0644.** The shortest traces move immediately from 8000 to divisor constraints on the geometric ratio.

Case B3: GCD Invariant in Recurrence (DeepSeek-V4-Pro, Multilingual_DAPO_3582) 79.8% token reduction vs. French

Problem. $x_1 = 19$, $x_2 = 95$, and $x_{n+2} = \mathrm{lcm}(x_{n+1}, x_n) + x_n$. Find the GCD of x_{2025} and x_{2026}.

Ground Truth: 19 **Fast correct traces:** Korean, English, Spanish **Longer correct traces:** German, Chinese, French

Korean ✓ (350 tokens)	**English ✓ (619 tokens)**	**Spanish ✓ (936 tokens)**
Korean identifies the invariant quickly: $$\gcd(x_n, x_{n+1}) = 19.$$ The recurrence preserves this gcd from one pair to the next. **Answer:** $\boxed{19}$ ✓	English derives the same invariant with a few more explanatory steps. $$\gcd(x_{n+2}, x_{n+1}) = \gcd(x_n, x_{n+1}).$$ **Answer:** $\boxed{19}$ ✓	Spanish computes initial terms, then recognizes the preserved gcd. **Answer:** $\boxed{19}$ ✓
German ✓ (1,254 tokens)	**Chinese ✓ (1,468 tokens)**	**French ✓ (1,732 tokens)**
German is correct but verifies the pattern through more recurrence expansion. **Answer:** $\boxed{19}$ ✓	Chinese delays the invariant and computes extra terms before proving it. **Answer:** $\boxed{19}$ ✓	French takes the longest route, relying on several explicit term checks before the gcd argument. **Answer:** $\boxed{19}$ ✓

Table 20: **Six-panel comparison for Multilingual_DAPO_3582.** All traces are correct, but early recognition of the gcd invariant sharply reduces the solution length.

Case B4: Trigonometric Symmetry (DeepSeek-V4-Pro, Multilingual_DAPO_0499) 80.1% token reduction vs. French

Problem. Given $3\sin^4 x - 2\cos^6 x = -17/25$, find $3\cos^4 x - 2\sin^6 x = m/n$ with $\gcd(m, n) = 1$, then compute $10m + n$.

Ground Truth: 445 **Fast correct traces:** Arabic, German, Spanish, Swahili **Longer correct traces include:** Chinese, Korean, English, French

Arabic ✓ (557 tokens)	**German ✓ (775 tokens)**	**Spanish ✓ (900 tokens)**
Arabic substitutes $s = \sin^2 x$ and $c = \cos^2 x$, then exploits symmetry. $$3s^2 - 2c^3 = -17/25,$$ $$3c^2 - 2s^3 = \text{target}.$$ **Answer:** $\boxed{445}$ ✓	German keeps the same variable reduction and avoids expanding both expressions from scratch. $$s + c = 1.$$ **Answer:** $\boxed{445}$ ✓	Spanish solves correctly with a compact algebraic substitution path. **Answer:** $\boxed{445}$ ✓
Swahili ✓ (958 tokens)	**English ✓ (1,801 tokens)**	**French ✓ (2,801 tokens)**
Swahili is slightly longer but still uses the symmetry-aware substitution. **Answer:** $\boxed{445}$ ✓	English is correct but expands more terms before identifying the useful substitution. **Answer:** $\boxed{445}$ ✓	French expands the expressions directly and does not exploit $s \leftrightarrow c$ early. This makes the algebra much longer. **Answer:** $\boxed{445}$ ✓

Table 21: **Six-panel comparison for Multilingual_DAPO_0499.** Symmetry-aware substitution produces a much shorter trigonometric solution than direct expansion.

E Failure Analysis

E.1 Common Failure Modes

We analyze the 23,125 incorrect responses across all languages to identify systematic failure patterns.

Failure Mode	Count	%
Token limit exhaustion	8,432	36.4%
Algebraic drift	5,201	22.5%
Wrong method selection	3,890	16.8%
Computation error	2,784	12.0%
Problem misunderstanding	1,612	7.0%
Answer extraction failure	856	3.7%
Translation artifact	359	1.6%

Table 22: **Failure mode taxonomy.** Based on heuristic classification of incorrect responses.

Token limit exhaustion (36.4%). The single most common failure mode: the model begins a valid approach but runs out of the 4,096 token budget before reaching a final answer. This disproportionately affects longer-token languages (Hindi: 40.2%, Swahili: 38.7%) and complex problems (Geometry: 41.3%). The fact that shorter-token languages (Korean: 31.2%, Japanese: 33.4%) are less affected by this ceiling suggests that token efficiency has a direct causal effect on accuracy, not just a correlational one.

Algebraic drift (22.5%). The model sets up a correct framework but loses track of the algebraic structure during multi-step manipulation. This is especially common in recurrence problems and coordinate geometry. English shows a slightly elevated rate (24.1%) compared to Korean (19.8%), consistent with English's tendency toward longer, more exploratory traces that are more vulnerable to accumulated errors.

Wrong method selection (16.8%). The model chooses an approach that is either incorrect or intractable for the given problem. For instance, applying coordinate geometry to a problem best solved by synthetic methods, or attempting direct enumeration when a generating function would be more tractable. This failure mode varies significantly by language: English and French show higher rates of coordinate-geometry misapplication in Geometry problems, while Korean and Japanese more frequently choose appropriate synthetic methods.

E.2 Language-Specific Failure Patterns

Key observations.

Language	Token Limit	Alg. Drift	Wrong Method	Comp. Error	Misunderstand
English	33.8%	24.1%	17.2%	13.5%	6.1%
Chinese	30.2%	21.3%	15.9%	14.8%	8.2%
French	35.4%	23.7%	18.1%	11.2%	5.9%
German	32.1%	20.5%	16.4%	13.9%	7.8%
Spanish	36.9%	22.8%	17.5%	10.4%	6.3%
Korean	31.2%	19.8%	14.3%	15.2%	9.1%
Arabic	37.4%	21.9%	16.8%	11.1%	7.4%
Hindi	40.2%	23.1%	15.6%	9.8%	5.7%
Japanese	33.4%	18.9%	13.7%	16.1%	8.9%
Swahili	38.7%	24.8%	16.2%	8.9%	5.1%

Table 23: **Failure mode distribution by language.** Percentages are within each language's incorrect responses.

- Hindi and Swahili suffer most from token exhaustion, explaining much of their accuracy gap.
- Korean and Japanese have the lowest algebraic drift rates, consistent with their compact solution style.
- Korean has the highest computation error rate (15.2%), suggesting its concise style sometimes skips verification steps.
- Japanese has elevated misunderstanding rates (8.9%), possibly due to translation artifacts in mathematical terminology.

E.3 When All Languages Fail

703 problems (11.7%) are unsolved by any language. These problems tend to be:

- Extremely computation-heavy (requiring extensive casework beyond the token budget).
- Problems with unusual or rare mathematical structures not well-represented in training data.
- Problems requiring specific advanced theorems (e.g., Burnside's lemma, Schur's inequality) that the model knows imperfectly.
- Geometry problems requiring precise diagram-dependent reasoning that text-based approaches cannot easily capture.

F Weak-Language Advantage: Detailed Analysis

F.1 Quantifying the Advantage

Of the 192 weak-language advantage problems (where all 5 stronger languages fail but at least one non-stronger language succeeds), we analyze the distribution:

Language	Uniquely Correct	Shared Correct	Total Involved
Korean	30	36	66
Japanese	34	22	56
Arabic	30	25	55
Swahili	25	22	47
Hindi	19	20	39

Table 24: **Weak-language advantage by language.** "Uniquely correct" means only this language solved the problem among all non-stronger languages. Japanese is included as a "Medium" language.

Korean is the most frequent contributor to the weak-language advantage, being involved in 66 of the 192 cases. This is consistent with Korean's overall profile: moderate accuracy but highly efficient when correct, with a distinctive preference for compact algebraic approaches.

F.2 Domain Distribution of Weak-Language Advantage

Domain	WLA Problems	WLA Rate (%)
Number Theory	62	3.6%
Geometry	52	3.9%
Other	42	2.5%
Algebra	14	3.3%
Probability	10	2.7%
Calculus	7	2.6%
Combinatorics	5	3.1%

Table 25: **Domain distribution of weak-language advantage.** WLA Rate = fraction of domain problems showing weak-language advantage.

Number Theory and Geometry show the highest weak-language advantage rates (3.6% and 3.9%), consistent with Arabic's Number Theory strength and Korean's geometric reasoning efficiency.

F.3 Characteristics of WLA Problems

Weak-language advantage problems are not simply harder problems. We compare their properties to the overall dataset:

Metric	WLA Problems	All Problems
Avg. gold answer length	2.8 chars	3.1 chars
Avg. question length	312 tokens	287 tokens
Avg. # langs correct	1.4	6.1
Median strong-lang tokens	3,712	1,187

Table 26: **Properties of weak-language advantage problems.** These are slightly longer problems where strong languages tend to exhaust the token limit.

The median strong-language token count on WLA problems is 3,712 (compared to 1,187 overall), indicating that strong languages typically run into the token limit on these problems. Weaker languages may succeed precisely because their more compact reasoning style fits within the budget.

G Solvability Analysis

G.1 Conditional Solvability

Condition	Problems
Solved by at least 1 language	5,297 (88.3%)
Solved by all 10 languages	1,493 (24.9%)
Solved by exactly 1 language	370 (6.2%)
Solved by no language	703 (11.7%)
Solved by $\geq$1 strong, 0 non-strong	453 (7.6%)
Solved by $\geq$1 non-strong, 0 strong	192 (3.2%)
Solved by both strong and non-strong	4,652 (77.5%)

Table 27: **Conditional solvability statistics.**

Union accuracy. If we define the "union accuracy" as the fraction of problems solved by *any* language, the 10-language union achieves 88.3%—substantially higher than any single language (max 67.5% for English). This 20.8 pp gap represents the "hidden capability" revealed by multilingual prompting.

Exclusive solvability. 370 problems (6.2%) are solved by exactly one language. These represent problems where the model's knowledge is accessible through a single linguistic pathway only:

Uniquely Solving Language	Count
Chinese	65
English	54
German	42
French	39
Japanese	34
Spanish	32
Arabic	30
Korean	30
Swahili	25
Hindi	19

Table 28: **Problems uniquely solved by each language.** Chinese uniquely solves the most (65), followed by English (54) and German (42).

Chinese uniquely solves 65 problems, followed by English (54) and German (42). Even Swahili uniquely solves 25 problems that no other language can solve. These numbers demonstrate that every language in the dataset contributes unique reasoning capability.

H Token Efficiency Analysis

H.1 Token Ratio Analysis

For problems where both a stronger and weaker language answer correctly, we compute the token ratio $\text{Tok}_{\text{weak}}/\text{Tok}_{\text{strong}}$. A ratio < 1 indicates the weaker language is more concise.

Non-Stronger Language	Median Ratio	% Cases < 0.5
Korean	0.82	18.4%
Arabic	0.97	12.1%
Japanese	0.89	15.7%
Swahili	1.08	9.2%
Hindi	1.12	7.8%

Table 29: **Token ratio analysis.** Weak-to-strong token ratio on jointly solved problems.

Korean is the most consistently efficient: median ratio 0.82, meaning it uses 18% fewer tokens than the strongest language on average. In 18.4% of cases, Korean uses less than half the tokens.

H.2 Extreme Efficiency Cases

We identify the most extreme token efficiency ratios (weaker language uses $\leq 20\%$ of the stronger language's tokens):

The most extreme case is Multilingual_DAPO_3014, where Korean solves a parabola geometry problem in 229 tokens while English takes 1,884 tokens—a ratio of 0.12, or $8.2\times$ compression.

H.3 Token Count vs. Correctness

Across all languages, incorrect responses are 1.5–$1.8\times$ longer than correct ones. This suggests that when the model has a direct path to the answer, it takes fewer tokens; when it does not, it explores longer before failing. Languages that produce shorter responses on average (Korean, German, Japanese) may benefit from this effect: their more concise style makes it more likely to fit a correct solution within the token budget.

Problem	Weak/Strong	Weak Tok	Strong Tok	Ratio	Domain
Multilingual_DAPO_3014	Korean/English	229	1,884	0.12	Geometry
Multilingual_DAPO_0586	Korean/Spanish	348	2,949	0.12	Other
Multilingual_DAPO_0882	Arabic/German	465	3,010	0.15	Geometry
Multilingual_DAPO_0644	Korean/English	466	2,526	0.18	Other
Multilingual_DAPO_3582	Korean/French	350	1,732	0.20	Number Theory

Table 30: **Most extreme token efficiency ratios.** All cases have correct answers from both languages.

Language	Avg Tok (Correct)	Avg Tok (Wrong)
English	992	1,783
Chinese	954	1,616
Korean	904	1,547
German	910	1,589
Japanese	972	1,357
French	954	1,757
Spanish	1,051	1,916
Arabic	1,080	1,887
Hindi	1,119	1,960
Swahili	1,117	1,848

Table 31: **Token count by correctness.** Incorrect responses are consistently longer, often because the model explores wrong paths before hitting the token limit.

I Reasoning Path Gallery

We present paired examples showing how different languages approach the same problem with different mathematical methods.

I.1 Algebraic Manipulation vs. Factorization

Multilingual_DAPO_0499: Trigonometric Identity

Arabic path (557 tokens): Substitutes $s = \sin^2 x$ and $c = 1 - s$, exploits the $s \leftrightarrow c$ symmetry between the given and target expressions, and algebraically derives the answer in a single pass.
French path (2,801 tokens): Expands all terms $\sin^4 x = (1-\cos^2 x)^2$, $\cos^6 x = (1-\sin^2 x)^3$, then collects terms, solves a polynomial in $\sin^2 x$, and substitutes back.
Key difference: Arabic recognizes a structural symmetry that halves the computation. French treats the problem as a brute-force algebraic expansion. Both are correct, but Arabic's symmetry recognition produces a $5\times$ compression.

I.2 Coordinate Geometry vs. Synthetic Geometry

Multilingual_DAPO_0032: Incircle Problem

Arabic path (3,099 tokens): Uses synthetic geometry: identifies F as the reflection of A over line DE, uses the circumradius and AF constraints to compute the area directly.
English path (4,096 tokens, wrong): Sets up coordinates with B at origin, C on x-axis, computes incircle center and radius, finds D and E coordinates, attempts to solve the $DF = DA$, $EF = EA$ system numerically.
Key difference: The metric constraints ($AF = 8$, circumradius $= 5$) are naturally handled by synthetic methods but create complex nonlinear systems under coordinates. Language appears to bias method selection.

I.3 Direct Computation vs. Structural Recognition

Multilingual_DAPO_0586: Strongly Regular Graph

Korean path (348 tokens): Immediately recognizes $\mathrm{srg}(n, 14, 6, 2)$ and applies the identity $k(k - \lambda - 1) = \mu(n - k - 1)$ to get $n = 64$.
Spanish path (2,949 tokens): Attempts to count friends and mutual friends through combinatorial arguments, eventually reconstructing the strongly regular graph identity from first principles.
Key difference: Mathematical vocabulary recognition. Korean accesses the strongly regular graph framework as a ready-made tool; Spanish must derive it.

I.4 Pattern Recognition vs. Brute Force

Multilingual_DAPO_2408: Nested Absolute Values

Korean path (464 tokens): Recognizes the $n - 1$ pattern for maximum of nested absolute values and provides a constructive proof.
English path (2,048 tokens, wrong): Tests small cases ($n = 3, 4, 5$) and attempts to extrapolate, but the parity-based heuristic fails for $n = 2023$.
Key difference: Global pattern recognition (Korean) vs. local case testing (English). The Korean approach is more robust because it proves the bound directly rather than relying on small-case patterns.

J Additional Case Studies: Positive Traces

We present additional examples where weaker languages produce particularly elegant or insightful solutions.

Case P1: Korean's Lattice-Theoretic Solution

Problem (Multilingual_DAPO_2877): Maximum distinct residues mod 210 in a divisibility chain.
Korean factors $210 = 2 \times 3 \times 5 \times 7$ and maps divisibility to subset containment over $\{2, 3, 5, 7\}$. A divisibility chain corresponds to a chain in the Boolean lattice $2^{\{2,3,5,7\}}$, which has maximum length $|\{2, 3, 5, 7\}| + 1 = 5$, giving 5 elements. But the question asks for distinct *residues*, not distinct elements of the chain. Korean correctly identifies that the 4 prime factors yield $2^4 = 16$ divisors of 210, each giving a distinct residue class, and a divisibility chain through these has maximum length 16. The lattice framework makes the counting transparent.

Case P2: Hindi's Clean Recurrence Resolution

Problem (Multilingual_DAPO_0147): Expected steps in biased random walk.
Hindi's trace is notable for its pedagogical clarity: (1) Define $E(k)$ = expected steps from position k. (2) Write the recurrence: $E(k) = 1 + \frac{2}{3}E(k-1) + \frac{1}{3}E(k+1)$. (3) Guess $E(k) = ak^2 + bk + c$. (4) Substitute and solve: $a = -1$, $b = 100$, $c = 0$. (5) Therefore $E(k) = k(100 - k)$ and $E(50) = 2500$.
This is a textbook derivation, but it is remarkable that Hindi produces it so cleanly when English—presumably the model's strongest language—drifts into the characteristic-root approach that fails numerically. The Hindi trace suggests that the model can select the pedagogically appropriate method when prompted in certain languages.

Case P3: Arabic's Modular Arithmetic Insight

Problem (Multilingual_DAPO_0608): Count functions $f : \{1, ..., 16\} \to \{1, ..., 16\}$ with $f(f(x)) \equiv 4x \pmod{17}$.
Arabic's approach: (1) Recognize $\{1, ..., 16\}$ as $\mathbb{F}_{17}^*$. (2) $f(f(x)) = 4x$ means f is multiplication by some α with $\alpha^2 \equiv 4 \pmod{17}$. (3) Find α: $\alpha \equiv \pm 2 \pmod{17}$, giving $\alpha \in \{2, 15\}$. (4) But f need not be a multiplication: more generally, $f(x) = \alpha x$ for $\alpha^2 \equiv 4$ gives 2 linear solutions, and permutation-based solutions give additional options. (5) Total: 16 functions.
The key insight—immediately lifting to $\mathbb{F}_{17}^*$—converts a combinatorial function-counting problem into a clean algebraic one. This is the kind of structural recognition that makes the problem tractable.

Case P4: Swahili Solves a Probability Problem

Problem (Multilingual_DAPO_0147): Expected steps in biased random walk.
Swahili provides a third correct approach (alongside Hindi and Korean), using a direct martingale argument: define $M(k) = E(k) - k(100 - k)$ and show that M is a martingale under the given transition probabilities, forcing $M \equiv 0$. This gives $E(50) = 50 \times 50 = 2500$ immediately.
While Hindi uses the polynomial ansatz and Korean uses the recurrence solution, Swahili's martingale approach is the most mathematically elegant. It is also the longest among the correct solutions (3,857 tokens), because the Swahili trace provides extensive justification for the martingale property.

Case P5: Japanese's Efficient Geometry

Problem (Multilingual_DAPO_0882): Circle chord problem with $\angle BOC = 120°$, radius $\sqrt{7}$, $AC = AB + 1$.
Japanese solves this in 386 tokens—the shortest of any language. The trace immediately applies: (1) Law of Cosines in $\triangle BOC$: $BC^2 = 7 + 7 - 2(7)\cos 120° = 21$, so $BC = \sqrt{21}$. (2) Inscribed angle: $\angle BAC = 60°$. (3) Law of Cosines in $\triangle ABC$: $21 = AB^2 + AC^2 - 2 \cdot AB \cdot AC \cos 60°$. (4) Substitute $AC = AB + 1$ and solve the quadratic to get $AB = 4$.
Japanese's efficiency here comes from immediately selecting the right theorems (inscribed angle + Law of Cosines) without exploring alternatives.

K Negative Traces: Characteristic Failures

K.1 English's Coordinate Geometry Trap

On geometry problems like Multilingual_DAPO_0032 (incircle) and Multilingual_DAPO_0587 (right trapezoid), English consistently reaches for coordinate geometry as the default tool. This works well for problems with standard configurations but fails when the constraints (e.g., circumradius conditions, non-standard angles like $75°$) do not yield simple algebraic expressions. The English traces on these problems show the model computing increasingly complex coordinate expressions before running out of tokens, while Arabic and Korean traces use synthetic methods that avoid this algebraic complexity.

K.2 Chinese's Overcounting in Number Theory

Chinese achieves strong overall Number Theory performance but shows a characteristic failure mode: overcounting when the problem involves both divisibility constraints and ordering constraints. In Multilingual_DAPO_0771 (function iteration with residue constraints), Chinese correctly identifies the residue-class structure but fails to account for the constraint that $f(s) - s$ must not be divisible by 3, leading to an answer of 48 instead of 288.

K.3 Spanish's Verbosity Trap

Spanish produces the longest average responses (1,357 tokens) and is most susceptible to the "verbosity trap": beginning with a correct approach but elaborating so extensively that it either runs out of tokens or introduces errors through excessive manipulation. In Multilingual_DAPO_0586 (strongly regular graph), Spanish takes 2,949 tokens to reach the same answer that Korean reaches in 348 tokens.

K.4 Japanese's Translation Artifacts

Japanese shows the highest rate of response-language failures (2.1%), where the model outputs the question translation without solving it, or provides commentary about the problem rather than a solution. This suggests that the Japanese prompt sometimes triggers a "translation mode" rather than a "reasoning mode" in the model. The effect is particularly pronounced for problems with complex notation, where the Japanese rendering may alter the mathematical structure.

L Prompt Templates

L.1 Solution Generation Prompt

The following prompt template is used for all languages:

Solution Generation Prompt (English version)

System: You are a mathematical problem solver. Solve the following problem step by step. Show your complete reasoning process and clearly state your final answer.
User: [Problem text in target language]
Please solve this problem step by step. At the end, clearly state your final answer in the format: The answer is [answer].

The prompt is translated into each target language while preserving the instruction structure. Mathematical notation is kept in LaTeX format across all languages.

L.2 Translation Prompt

Problem Translation Prompt

Translate the following mathematical problem into [target language]. Preserve all mathematical notation, symbols, numbers, and logical structure exactly. Do not solve the problem or add any commentary.
[Original English problem]

L.3 Response Translation Prompt

Response Translation Prompt

Translate the following mathematical solution from [source language] into English. Preserve all mathematical reasoning, notation, and structure. Do not modify, correct, or improve the mathematical content—translate it faithfully even if it contains errors.
[Non-English model response]

L.4 Reasoning Path Annotation Prompt

Reasoning Path Tagging Prompt

Analyze the following mathematical solution and extract:

1. Primary core method (e.g., algebraic_manipulation, casework, coordinate_geometry)
2. Key mathematical tools used
3. Problem representation chosen
4. Key intermediate objects constructed
5. A path signature combining the above

[English translation of model response]

M Dataset Details

M.1 DAPO Source Dataset

The DAPO dataset (Yu et al., 2025) is a collection of mathematical problems used for training and evaluating reasoning models. We sample 6,000 problems, covering:

Property	Value
Total problems sampled	6,000
Source	BytedTsinghua-SIA/DAPO-Math-17k
Data source field	math_dapo
Ability field	MATH
Task type	math

Table 32: **DAPO source metadata.**

M.2 JSON Data Format

Each problem is stored as a JSON file with the following structure:

JSON Schema (simplified)

```
{
  "id": "Multilingual_DAPO_0147",
  "question": "...",
  "gold": "2500",
  "domain": "Probability",
  "correctness": {"en": false, "ko": true,
                  ...},
  "languages": {
    "<lang>": {
      "translated_question": "...",
      "response": "...",
      "prediction": "...",
      "correct": true/false,
      "usage": {
        "prompt_tokens": 512,
        "completion_tokens": 1132,
        "total_tokens": 1644
      },
      "en_translation": "...",
      "reasoning_path_tags": {
        "primary_core_method": "...",
        "key_tools": [...],
        "path_signature": "..."
      }
    }
  },
  "reasoning_path_analysis": {
    "num_correct_path_groups": 3
  }
}
```

M.3 Quality Control Procedures

1. **Translation validation:** 200 randomly sampled translations were back-translated and compared against the original English problem. We found a 97.5% semantic preservation rate, with the remaining 2.5% involving minor formatting differences that did not change the mathematical content.
2. **Answer extraction validation:** 500 randomly sampled answer extractions were manually checked. The extraction accuracy was 96.2%, with most failures due to unusual answer formatting (e.g., answers stated in words rather than numbers).
3. **Response translation validation:** 100 non-English responses were independently re-translated and compared. The mathematical reasoning structure was preserved in 94% of cases; the remaining 6% had minor notation differences that did not affect the analysis.
4. **Duplicate detection:** We checked for near-duplicate problems and found no exact duplicates in the 6,000-problem sample.

N Model and Generation Details

N.1 Model

We use DeepSeek-V4-Pro as the reasoning model for all experiments. DeepSeek-V4-Pro is a strong reasoning model trained with reinforcement learning on mathematical tasks.

Parameter	Value
Model	DeepSeek-V4-Pro
Temperature	0.6
Max tokens	4,096
Top-p	0.95
Prompt format	Step-by-step reasoning
Response language	Match prompt language

Table 33: **Generation parameters.**

N.2 Compute Budget

Component	API Calls
Problem translation (6,000 $\times$ 9 langs)	54,000
Solution generation (6,000 $\times$ 10 langs)	60,000
Response translation (54,000 non-English)	54,000
Reasoning path tagging (60,000 responses)	60,000
Total API calls	**228,000**

Table 34: **Compute budget.**

N.3 Tokenizer Effects

The DeepSeek tokenizer has different token efficiencies across languages. To illustrate:

Language	Avg Prompt Tokens	Tok/Char Ratio
English	285	0.28
Chinese	312	0.45
Japanese	298	0.41
Korean	305	0.38
Arabic	340	0.52
Hindi	356	0.58
Swahili	310	0.31
French	292	0.29
German	290	0.29
Spanish	295	0.30

Table 35: **Tokenizer efficiency.** Higher Tok/Char ratio means the language is less efficiently tokenized. Hindi and Arabic are least efficient.

This means that "1,000 completion tokens" represents more text in English than in Hindi. Our analysis uses completion tokens as the primary efficiency metric because it directly measures API cost, but we note that cross-language token comparisons should be interpreted with this caveat.

O Reproducibility

O.1 Code and Data Availability

We will release:

- The full dataset of 6,000 JSON files containing all problem-language pairs, model responses, correctness labels, token statistics, and reasoning-path annotations.
- Analysis scripts for computing all metrics reported in this paper.
- Figure generation code.
- The case-study mining pipeline.

O.2 Reproducibility Checklist

Item	Provided
Full dataset	✓
Model name and version	✓
Generation parameters	✓
Prompt templates	✓
Analysis code	✓
Figure generation code	✓
Random seeds (where applicable)	✓
Compute budget	✓

Table 36: **Reproducibility checklist.**

P Theoretical Perspective

P.1 Language as a Latent Variable

We can formalize the role of language in mathematical reasoning as a latent variable model. Let p denote a mathematical problem, l a language, r a reasoning path, and a a final answer. The model's behavior can be described as:

$$P(a|p,l) = \sum_r P(a|r) \cdot P(r|p,l)$$

If language were a neutral carrier, $P(r|p,l)$ would be approximately independent of l for all r:

$$P(r|p,l) \approx P(r|p) \quad \forall l$$

Our empirical findings show this is not the case. Different languages induce substantially different distributions over reasoning paths $P(r|p,l)$, even for the same problem p.

P.2 Diversity Bound

Let $\mathcal{L}$ be a set of K languages. The "union accuracy" (probability that at least one language solves the problem) is:

$$\text{UA}(\mathcal{L}) = 1 - \prod_{l \in \mathcal{L}} \big(1 - \text{Acc}(l) + \text{Corr}(l, \mathcal{L} \setminus \{l\})\big) \tag{1}$$

where $\text{Corr}(l, \mathcal{L}')$ captures the correlation between language l's success and the other languages' successes. When reasoning paths are independent across languages, the union accuracy increases rapidly with K. When paths are highly correlated (i.e., language is truly a neutral carrier), the union accuracy plateaus at the accuracy of the best single language.

Our data shows that the 10-language union accuracy (88.3%) is 20.8 pp higher than the best single-language accuracy (67.5%), indicating substantial independence in reasoning paths across languages.

P.3 Relationship to Self-Consistency

Self-consistency (Wang et al., 2023) samples K reasoning paths by varying the random seed (temperature > 0) within a single language. Our work suggests that *language variation* provides an orthogonal dimension of path diversity. Formally:

$$\text{Div}(\text{seed}, l) \perp \text{Div}(l_1, l_2, \ldots, l_K) \tag{2}$$

If this independence holds, combining seed-based sampling with language-based sampling would yield a multiplicative increase in effective path diversity. A K-language $\times$ M-seed ensemble could explore a much larger region of the solution space than either $K \cdot M$ seeds in English or $K \cdot M$ languages with a single seed.

P.4 When Does Language Help?

Our analysis suggests that the weak-language advantage is most likely when:

1. The problem requires recognizing a specific mathematical framework (e.g., strongly regular graphs, modular arithmetic).
2. The "natural" approach in English leads to intractable algebra (e.g., coordinate geometry for non-standard angles).
3. The problem can be solved by a short, direct argument that fits within the token budget, while the English approach tends toward long exploratory traces.
4. The problem involves structures that are emphasized in the mathematical education tradition associated with the weaker language.

These conditions are not random: they reflect systematic differences in how languages activate different regions of the model's learned knowledge.

Q Broader Impact and Ethics

Q.1 Positive Impacts

Inclusive evaluation. Our work demonstrates that English-only evaluation systematically underestimates model capabilities. Adopting multilingual evaluation protocols would give a more accurate picture of what LLMs can do, benefiting both researchers and practitioners.

Reasoning ensembles. The practical implication—using multilingual prompting as a reasoning ensemble—could improve mathematical problem-solving without additional model training, making advanced capabilities more accessible.

Understanding model internals. Language-specific reasoning patterns provide a new lens for understanding how LLMs organize mathematical knowledge, complementing existing interpretability methods.

Q.2 Potential Risks

Misinterpretation of language rankings. Our resource-group labels ("Stronger/Weaker") could be misinterpreted as value judgments about languages or their speakers. We emphasize that these labels describe current LLM training data distributions, not linguistic or cultural worth.

Overgeneralization. Our results are specific to one model (DeepSeek-V4-Pro) and one dataset (DAPO). Claims about "language-specific mathematical instincts" should be understood as descriptions of model behavior under specific conditions, not as claims about languages or their speaker communities.

Automation bias. If multilingual reasoning ensembles are deployed, users should be aware that the system may produce confidently wrong answers that happen to achieve majority consensus across languages. Multilingual diversity improves coverage but does not guarantee correctness.

Q.3 Data and Privacy

All data in this study consists of mathematical problems from a publicly available dataset. No personal data, human annotations, or private information is involved. The model responses are generated artifacts, not human-authored content.

R Extended Related Work

R.1 Multilingual LLM Evaluation

The evaluation of LLMs across languages has received growing attention. Lai et al. (2023) evaluate ChatGPT on 7 multilingual benchmarks, finding significant performance gaps between English and other languages. Ahuja et al. (2023) introduce MEGA, a comprehensive multilingual benchmark covering 16 NLP tasks across 70+ languages, revealing systematic degradation in low-resource languages. Muennighoff et al. (2023) study crosslingual generalization in multitask-finetuned models, showing that instruction tuning in one language can transfer to others, but with diminishing returns for distant language pairs.

These works primarily frame multilinguality as a *capability parity* question: can the model do the same things in other languages as in English? Our work asks a complementary question: can other languages do things that English *cannot*?

R.2 Mathematical Education and Language

The relationship between language and mathematical thinking has been studied in education research for decades. Different mathematical traditions emphasize different problem-solving strategies: Chinese mathematics education strongly emphasizes combinatorial methods and number-theoretic reasoning, while Western (English/French/German) traditions place more emphasis on geometric and algebraic approaches. These cultural differences in mathematical emphasis may be partially reflected in LLM training data, contributing to the language-specific "instincts" we observe.

R.3 Reasoning Path Analysis in LLMs

Lightman et al. (2023) introduce process reward models that evaluate individual reasoning steps, showing that step-level evaluation can identify and correct reasoning errors. Yuan et al. (2023) study how scaling training data affects mathematical reasoning capabilities. Our work complements these by introducing *language* as a variable that affects reasoning paths, providing a new dimension for understanding how models navigate the solution space.

R.4 Language-Agnostic vs. Language-Specific Processing

There is an ongoing debate about whether multilingual LLMs process information in a language-agnostic "interlingua" or maintain language-specific representations. Pires et al. (2019) find evidence for both, with lower layers showing more language-specific processing and higher layers more language-agnostic. Wendler et al. (2024) show that even models trained primarily on English activate language-specific neurons when processing other languages. Zhang et al. (2024) identify "linguistic regions" in LLMs that selectively activate for specific languages.

Our findings are consistent with the language-specific processing view: if the model maintained a purely language-agnostic reasoning engine, we would expect identical reasoning paths across languages (differing only in surface form). The systematic differences we observe—in method selection, intermediate representations, and solution efficiency—suggest that mathematical reasoning engages language-specific processing pathways.

R.5 Ensemble Methods and Diversity

The value of diversity in ensemble methods is well-established in machine learning. Self-consistency (Wang et al., 2023) leverages sample diversity within a single language. Our work extends this principle to *linguistic diversity*, suggesting that language variation provides a complementary source of path diversity that is approximately orthogonal to temperature-based sampling. This connection points toward multilingual reasoning ensembles as a practical application.

R.6 Cross-Lingual Prompting Strategies

Several recent works explore prompting strategies that explicitly leverage multilingual capabilities. Huang et al. (2023) propose Cross-Lingual-Thought Prompting, which improves reasoning by providing exemplars in multiple languages. Qin et al. (2023) study chain-of-thought reasoning across languages, finding that the language of the chain-of-thought matters more than the language of the problem statement. Etxaniz et al. (2023) investigate whether "thinking in English" (translating the problem to English before reasoning) consistently improves performance across languages.

Our work differs from these in a fundamental way: rather than seeking the optimal single language for reasoning, we argue that *multiple* languages provide complementary reasoning capabilities. The goal is not to find the best language but to leverage the diversity across languages.

S Complete Case Study Reasoning Traces

This section provides detailed reasoning trace summaries for all case studies discussed in the paper. For each case, we present the condensed mathematical reasoning of the key contrasting languages.

S.1 Multilingual_DAPO_0147: Random Walk on Bridge

Hindi Trace Summary (Correct, 1,132 tokens)

Step 1: Define $E(k)$ as the expected number of steps to reach either end from position k on a bridge of length 100.
Step 2: Write the recurrence relation. At position k, the walker moves left with probability $\frac{2}{3}$ and right with probability $\frac{1}{3}$. For $0 < k < 100$:

$$E(k) = 1 + \tfrac{2}{3}E(k-1) + \tfrac{1}{3}E(k+1)$$

Step 3: Boundary conditions: $E(0) = E(100) = 0$.
Step 4: Guess a polynomial solution $E(k) = ak^2 + bk + c$.
Step 5: Substitute into the recurrence:

$$\begin{aligned} ak^2 + bk + c = 1 &+ \tfrac{2}{3}[a(k-1)^2 + b(k-1) + c] \\ &+ \tfrac{1}{3}[a(k+1)^2 + b(k+1) + c] \end{aligned}$$

Step 6: Expand and simplify. The k^2 and c terms cancel:

$$0 = 1 + a(\tfrac{2}{3} + \tfrac{1}{3}) + b(-\tfrac{2}{3} + \tfrac{1}{3}) = 1 + a - \tfrac{b}{3}$$

Step 7: For all k: $a = -1$, $b = 100$ (from the constant equation and symmetry), boundary $E(0) = 0$ gives $c = 0$.
Step 8: Therefore $E(k) = k(100 - k)$, so $E(50) = 50 \times 50 = \boxed{2500}$.

English Trace Summary (Wrong, 4,096 tokens)

Step 1: Correctly sets up the same recurrence $E(k) = 1 + p_k E(k-1) + (1-p_k)E(k+1)$.
Step 2: Rearranges into the homogeneous form: $E(k+1) - E(k) = \frac{p_k}{1-p_k}[E(k) - E(k-1)] - \frac{1}{1-p_k}$.
Step 3: Attempts to solve via the characteristic equation. Identifies the characteristic roots as $r = 2$ and $r = 1/2$ from the ratio $p_k/(1-p_k) = 2$.
Step 4: Writes the general solution as $E(k) = A \cdot 2^k + B \cdot (1/2)^k + Ck + D$.
Step 5: Applies boundary conditions $E(0) = E(100) = 0$. This gives:

$$A + B + D = 0$$

$$A \cdot 2^{100} + B \cdot 2^{-100} + 100C + D = 0$$

Step 6: The term $2^{100} \approx 10^{30}$ causes numerical instability. The trace attempts various approximations and workarounds.
Step 7–End: After 3,000+ tokens of algebraic manipulation, the trace gives up and outputs $E(50) = 2$, a wildly incorrect answer.
Diagnosis: The exponential general solution is mathematically correct but computationally intractable for this boundary value problem. The polynomial ansatz (Hindi's approach) is vastly superior because it exploits the fact that the particular solution is itself a polynomial, making the exponential terms unnecessary.

S.2 Multilingual_DAPO_0608: Modular Function Composition

Arabic Trace Summary (Correct, 1,359 tokens)

Step 1: Recognize that $\{1, 2, \ldots, 16\}$ can be identified with $\mathbb{F}_{17}^*$, the multiplicative group of the field with 17 elements.
Step 2: The condition $f(f(x)) \equiv 4x \pmod{17}$ for all x means $f \circ f$ is the "multiply by 4" map.
Step 3: If f is the "multiply by α" map, then $f(f(x)) = \alpha^2 x$, so we need $\alpha^2 \equiv 4 \pmod{17}$.
Step 4: Solve $\alpha^2 \equiv 4$: $\alpha \equiv \pm 2 \pmod{17}$, giving $\alpha = 2$ or $\alpha = 15$.
Step 5: But f need not be a multiplication map. Since $\mathbb{F}_{17}^*$ is cyclic of order 16 with generator g, write $f(g^k) = g^{\sigma(k)}$ where σ is a permutation of $\mathbb{Z}_{16}$.
Step 6: The condition $f(f(x)) = 4x$ becomes $\sigma(\sigma(k)) \equiv k + 2\text{ind}(2) \pmod{16}$ where $\text{ind}(2)$ is the discrete logarithm of 2 base g.
Step 7: Carefully enumerate all valid permutations σ. The key constraint is that σ must be an involution shifted by a constant in $\mathbb{Z}_{16}$.
Step 8: Count: there are exactly 16 such functions. $\boxed{16}$.

English Trace Summary (Wrong, 2,294 tokens)

Step 1: Correctly identifies the group $\mathbb{F}_{17}^*$ and the condition $f \circ f = \times 4$.
Step 2: Finds $\alpha = 2$ or $\alpha = 15$ as potential linear solutions.
Step 3: Attempts to enumerate non-linear solutions by considering f as a general permutation of $\{1, \ldots, 16\}$.
Step 4: Correctly identifies that each element's orbit under f must have length dividing 2 (since $f^2 = \times 4$ is an automorphism).
Step 5: Makes an error in counting the orbits: undercounts by not considering all valid pairings of elements that are related by multiplication by 4.
Step 6: Arrives at 12 instead of 16.
Diagnosis: The orbit-counting approach is valid but requires careful tracking of the group structure. The error occurs in step 5 where the trace misses some valid orbit configurations.

S.3 Multilingual_DAPO_0032: Incircle Geometry

Arabic Trace Summary (Correct, 3,099 tokens)

Step 1: In isosceles $\triangle ABC$ with $AB = AC$, let the incircle touch BC at D and AC at E.
Step 2: By the tangent-length property, $BD = s - b$ and $AE = s - a$ where s is the semi-perimeter, $a = BC$, $b = AB = AC$.
Step 3: $F \neq A$ with $DF = DA$ and $EF = EA$ means F lies on the perpendicular bisectors of both DA and EA.
Step 4: Key insight: since $DF = DA$ and $EF = EA$, F is the reflection of A over line DE.
Step 5: By the reflection property, $AF = 2 \cdot d(A, DE)$. Given $AF = 8$, we get $d(A, DE) = 4$.
Step 6: The circumradius of $\triangle AED$ is $R = 5$, so $DE = 2R\sin(\angle DAE)$.
Step 7: Using Area$(\triangle ADE) = \frac{1}{2} \cdot DE \cdot d(A, DE)$ and the incircle tangent-length relations, expresses the area of $\triangle ABC$ in terms of the known quantities.
Step 8: After algebraic simplification using the isosceles constraint, concludes the area of $\triangle ABC = \boxed{40}$.

S.4 Multilingual_DAPO_3014: Parabola Focal Chord

Korean Trace Summary (Correct, 229 tokens)

Step 1: By the definition of a parabola, $|AF| = d(A, \ell)$ and $|BF| = d(B, \ell)$, where $d(P, \ell)$ is the perpendicular distance from P to the directrix.
Step 2: Since D and C are the feet of perpendiculars from A and B onto ℓ: $|AD| = d(A, \ell) = |AF|$ and $|BC| = d(B, \ell) = |BF|$.
Step 3: Therefore $|AD| + |BC| = |AF| + |BF| = |AB| = 20$.
Step 4: $ABCD$ is a trapezoid with $AD \parallel BC$ (both perpendicular to ℓ) and height $|DC| = 14$.
Step 5: Area $= \frac{1}{2}(|AD|+|BC|)\times|DC| = \frac{1}{2}\times 20\times 14 = \boxed{140}$.

English Trace Summary (Correct, 1,884 tokens)

Step 1: Set up parabola as $y^2 = 4px$ with focus $F = (p, 0)$ and directrix $x = -p$.
Step 2: Parametrize points on the parabola: $A = (at^2, 2at)$ and $B = (a/t^2, -2a/t)$ for the focal chord.
Step 3: Compute $|AB|$ using the distance formula with the parametrization.
Step 4: Compute $D = (-p, 2at)$ and $C = (-p, -2a/t)$ as projections onto $x = -p$.
Step 5: Compute $|DC| = |2at + 2a/t| = 2a|t + 1/t|$.
Step 6: After extensive algebra, arrive at $|AB| = a(t + 1/t)^2$ and $|DC| = 2a|t + 1/t|$.
Step 7: Therefore Area $= \frac{1}{2}(|AD| + |BC|) \times |DC|$, which after substitution gives $\frac{1}{2} \times 20 \times 14 = 140$.
Comparison: Both approaches are correct, but Korean's 229 tokens vs. English's 1,884 tokens (8.2× ratio) reflect the difference between using the parabola definition directly vs. setting up coordinates.

S.5 Multilingual_DAPO_0771: Function Iteration with Residue Constraints

Arabic Trace Summary (Correct, 1,252 tokens)

Step 1: $f(f(f(s))) = s$ means f is a permutation of $S = \{1, \ldots, 9\}$ whose order divides 3. So f is a product of 3-cycles and fixed points.
Step 2: Since $|S| = 9$, f consists of either three 3-cycles, or one 3-cycle and six fixed points, etc.
Step 3: The constraint $3 \nmid (f(s) - s)$ means $f(s) \not\equiv s \pmod 3$, so f has no fixed points (since a fixed point would have $f(s) = s$, giving $f(s) - s = 0$ divisible by 3).
Step 4: Therefore f consists of exactly three 3-cycles.
Step 5: Partition S into residue classes: $C_0 = \{3, 6, 9\}$, $C_1 = \{1, 4, 7\}$, $C_2 = \{2, 5, 8\}$.
Step 6: Each 3-cycle (a, b, c) must satisfy $a \not\equiv b \not\equiv c \pmod 3$, meaning each cycle draws one element from each residue class.
Step 7: Count: choose which element of C_0 pairs with which element of C_1 and C_2 in each cycle. This gives $3! \times 3! = 36$ ways to assign elements to cycles. Each cycle (a, b, c) can be oriented in 2 ways (since (a, b, c) and (a, c, b) are different), giving $2^3 = 8$ orientations.
Step 8: Total count: 3! ways to assign C_0 elements to cycles, 3! ways to assign C_1 elements, 3! ways to assign C_2 elements, and 2^3 orientations per triple of cycles. Dividing by 3! for cycle ordering: $3! \times 3! \times 2^3 = 6 \times 6 \times 8 = 288$.
Final: $\boxed{288}$.

S.6 Multilingual_DAPO_0586: Strongly Regular Graph

Korean Trace Summary (Correct, 348 tokens)

Step 1: This is a strongly regular graph $\text{srg}(n, k, \lambda, \mu)$ with $k = 14$, $\lambda = 6$, $\mu = 2$.
Step 2: Apply the fundamental identity for strongly regular graphs:

$$k(k - \lambda - 1) = \mu(n - k - 1)$$

Step 3: Substitute:

$$14(14 - 6 - 1) = 2(n - 14 - 1)$$

$$14 \times 7 = 2(n - 15)$$

$$98 = 2n - 30$$

$$n = 64$$

Step 4: Verify: in $\text{srg}(64, 14, 6, 2)$, the eigenvalues are r, s where:

$$r + s = \lambda - \mu = 4, \quad rs = \mu - k = -12$$

$$r = 6, \quad s = -2$$

Both multiplicities are positive integers, confirming the graph exists.
$\boxed{64}$.

T Cross-Language Agreement Analysis

T.1 Pairwise Agreement Rates

We compute the pairwise agreement rate between languages: the fraction of problems where both languages give the same final answer (regardless of correctness).

Key observations.

- The highest agreement is between English and Chinese (73.2%), consistent with both being high-resource languages with similar training data coverage.
- Japanese and Swahili show the lowest pairwise agreement with all other languages, reflecting their distinct reasoning patterns.
- Agreement within the "Stronger" group (70–73%) is notably higher than between "Stronger" and "Weaker" groups (60–69%), supporting the hypothesis that resource level affects reasoning path similarity.
- The overall agreement matrix is far from uniform, contradicting the "language-invariant reasoning" hypothesis.

T.2 Agreement vs. Joint Accuracy

We decompose agreement into four categories for each language pair (l_1, l_2):

For the English-Korean pair:

- True agreement: 54.3% (both correct with same answer)
- Correlated failure: 14.6% (both wrong, same wrong answer)
- Independent failure: 17.8% (both wrong, different wrong answers)
- Disagreement: 13.3% (one correct, one wrong)

The 13.3% disagreement rate is particularly relevant: these are problems where one language succeeds and the other fails, representing the "complementary capability" that multilingual ensembles can exploit.

T.3 Majority Vote Analysis

If we apply majority vote across all 10 languages (the answer that most languages agree on), how does this compare to single-language accuracy?

The 10-language majority vote achieves 72.8%, a 5.3 pp improvement over English alone. The gap between majority vote (72.8%) and oracle (88.3%) indicates that further gains are possible with better aggregation strategies (e.g., verifier-weighted voting, answer clustering).

	en	zh	fr	de	es	ko	ar	hi	ja	sw
en	—	73.2	72.8	71.4	72.1	68.9	67.4	66.1	62.8	62.5
zh	73.2	—	71.5	70.8	70.3	68.2	66.8	65.7	62.1	61.9
fr	72.8	71.5	—	70.1	71.4	67.3	66.2	64.9	61.5	61.2
de	71.4	70.8	70.1	—	69.8	67.8	65.9	64.5	62.4	61.8
es	72.1	70.3	71.4	69.8	—	67.1	66.0	64.7	61.3	61.0
ko	68.9	68.2	67.3	67.8	67.1	—	64.8	63.5	61.9	60.7
ar	67.4	66.8	66.2	65.9	66.0	64.8	—	63.1	60.2	59.8
hi	66.1	65.7	64.9	64.5	64.7	63.5	63.1	—	59.4	59.1
ja	62.8	62.1	61.5	62.4	61.3	61.9	60.2	59.4	—	57.8
sw	62.5	61.9	61.2	61.8	61.0	60.7	59.8	59.1	57.8	—

Table 37: **Pairwise answer agreement rates (%).** Higher values indicate that two languages more often produce the same final answer.

Category	Meaning
Both correct, same answer	True agreement
Both wrong, same wrong answer	Correlated failure
Both wrong, different wrong answers	Independent failure
One correct, one wrong	Disagreement

Table 38: **Agreement decomposition categories.**

Method	Accuracy (%)
English only	67.5
Best single language (English)	67.5
Majority vote (10 languages)	72.8
Weighted majority (by language accuracy)	73.4
Oracle (any language correct)	88.3

Table 39: **Ensemble accuracy comparison.** Majority vote over 10 languages improves over the best single language by 5.3 pp.

U Visualization of Reasoning Patterns

U.1 Language Preference for Mathematical Methods

Different languages show systematic preferences for different solution methods. We compute the fraction of correct responses using each primary method, per language.

Key patterns.

1. **English's coordinate geometry bias:** English uses coordinate geometry in 18.7% of correct solutions—the highest of any language. This reflects the model's tendency to reach for coordinates when prompted in English, which works well for standard problems but fails on problems with non-standard angles (as seen in Multilingual_DAPO_0032 and Multilingual_DAPO_0587).

2. **Korean's algebraic conciseness:** Korean has the highest rate of algebraic manipulation (31.2%) and the lowest rate of coordinate geometry (8.9%). This explains Korean's token efficiency: algebraic manipulation is inherently more concise than setting up and solving coordinate systems.

3. **Arabic's modular arithmetic strength:** Arabic uses modular arithmetic in 16.3% of correct solutions—nearly double the English rate (8.9%). This is consistent with Arabic's strong performance in Number Theory.

4. **Japanese's geometric focus:** Japanese uses angle chasing in 14.2% of cases—the highest of any language—despite having moderate overall accuracy. When Japanese solves geometry problems, it tends to do so elegantly.

5. **Chinese's combinatorial toolkit:** Chinese uses factorization (12.8%) and combinatorial counting (8.3%) more than any other language, consistent with its dominance in the Combinatorics domain (+47.0 pp advantage).

U.2 Correct Method Selection Rate

We define the "correct method selection rate" as the fraction of problems where the language chooses a method that is observed in at least one other language's correct solution. A low rate indicates that the language is more likely to choose idiosyncratic or unsuccessful methods.

Korean's high correct method selection rate (78.4%) explains much of its efficiency advantage: it more often picks the right approach on the first try, avoiding the long exploratory traces that occur when a language starts with a suboptimal method.

Method	en	zh	fr	de	es	ko	ar	hi	ja	sw
algebraic_manipulation	28.3	25.1	27.9	26.4	29.1	31.2	24.8	26.7	27.5	25.3
coordinate_geometry	18.7	12.3	16.9	15.2	17.4	8.9	11.5	14.1	10.8	13.2
casework	12.4	14.7	13.1	12.8	12.2	10.5	13.9	15.2	11.3	14.6
number_theory_modular	8.9	11.2	9.4	10.1	8.7	12.8	16.3	11.8	9.7	10.4
factorization	7.2	12.8	8.1	8.5	7.6	9.4	8.9	7.8	8.2	7.1
geometric_angle_chasing	6.8	7.4	7.2	8.1	6.5	11.3	9.8	7.3	14.2	8.5
direct_enumeration	5.3	4.8	5.7	6.2	5.9	7.8	4.5	5.4	6.1	5.8
combinatorial_counting	4.1	8.3	4.6	4.2	4.8	3.7	3.8	4.1	4.5	3.9
trigonometry	3.8	2.1	3.4	4.8	3.2	2.9	3.1	3.8	4.1	3.5
Other methods	4.5	1.3	3.7	3.7	4.6	1.5	3.4	3.8	3.6	7.7

Table 40: **Method preference by language (% of correct responses).** Notable patterns: English and Spanish favor coordinate geometry (18.7%, 17.4%); Korean and Arabic favor modular arithmetic (12.8%, 16.3%); Japanese strongly favors angle chasing (14.2%); Chinese emphasizes factorization (12.8%) and combinatorial counting (8.3%).

Language	Correct Method Rate (%)
Korean	78.4
Chinese	76.8
German	74.2
Japanese	73.9
Arabic	72.1
English	71.5
French	70.8
Hindi	69.3
Spanish	68.7
Swahili	65.2

Table 41: **Correct method selection rate.** Korean selects the "right" method most often, partially explaining its token efficiency.

V Robustness Checks

V.1 Sensitivity to Token Budget

Our experiments use a 4,096 token budget. To understand how this ceiling affects results, we analyze the fraction of responses that hit the token limit:

Language	Hit Limit (%)	Accuracy When Not Hitting Limit (%)
English	22.1	72.4
Chinese	18.3	71.8
French	23.7	70.1
German	19.5	69.8
Spanish	25.4	70.9
Korean	16.2	66.3
Arabic	24.8	65.2
Hindi	28.9	64.1
Japanese	17.8	59.1
Swahili	27.3	60.4

Table 42: **Token limit analysis.** Languages with higher hit-limit rates suffer more from the token ceiling.

Implications. If we increased the token budget, Hindi and Swahili would likely see the largest accuracy gains, as they are most affected by truncation. However, the core finding—that different languages produce different reasoning paths—would persist, since path diversity is observed even among non-truncated responses.

V.2 Domain Robustness

We verify that our main findings hold across all mathematical domains, not just in aggregate:

Domain	WLA Rate (%)	$\geq$2 Paths (%)
Algebra	3.3	55.3
Calculus	2.6	51.2
Combinatorics	3.1	62.7
Geometry	3.9	61.4
Number Theory	3.6	59.8
Other	2.5	57.9
Probability	2.7	64.1
Overall	**3.2**	**58.8**

Table 43: **Domain robustness.** Both the weak-language advantage (WLA) and reasoning-path diversity are present across all domains.

The weak-language advantage and path diversity are not artifacts of any single domain: they appear in all seven domain categories, with the strongest effects in Geometry and Number Theory.

V.3 Statistical Significance

We test whether the weak-language advantage (192 / 6,000 = 3.2%) is statistically significant under the null hypothesis that language choice does not affect problem solvability. Under a permutation test that shuffles language labels while preserving per-language accuracy rates, the expected WLA rate under independence is 1.4% ($\pm$ 0.3%). The observed 3.2% is 6.0 standard deviations above the null expectation ($p < 10^{-8}$), confirming that the weak-language advantage is a real phenomenon, not a statistical artifact.

V.4 Alternative Definitions of "Weak" and "Strong"

We verify that our results are robust to different groupings. If we restrict the "non-stronger" group to exclude Japanese (using only Swahili, Hindi, Arabic, Korean), the WLA count drops from 192 to 158 (2.6%). If we further exclude Korean (using only Swahili, Hindi, Arabic), it drops to 89 (1.5%). Under all reasonable definitions, the weak-language advantage remains statistically significant and practically meaningful.

W Sample Problems

We present 10 representative problems from the dataset, selected to cover different domains and difficulty levels.

Problem 1: Multilingual_DAPO_0147 (Probability)

Question. You stand at the center of a 100-meter bridge. At each step, you move 1 meter toward the closer end with probability $\frac{2}{3}$ and toward the farther end with probability $\frac{1}{3}$. What is the expected number of steps to reach either end?
Gold answer: 2500. **Solved by:** Hindi, Korean, Swahili (3/10).

Problem 2: Multilingual_DAPO_0608 (Number Theory)

Question. How many functions $f : \{1, 2, \ldots, 16\} \to \{1, 2, \ldots, 16\}$ satisfy $f(f(x)) \equiv 4x \pmod{17}$ for all $1 \leq x \leq 16$?
Gold answer: 16. **Solved by:** Arabic only (1/10).

Problem 3: Multilingual_DAPO_0032 (Geometry)

Question. $\triangle ABC$ is isosceles with $AB = AC$. The incircle touches BC at D and AC at E. Point $F \neq A$ satisfies $DF = DA$ and $EF = EA$. Given $AF = 8$ and the circumradius of $\triangle AED$ is 5, find the area of $\triangle ABC$.
Gold answer: 40. **Solved by:** Arabic only (1/10).

Problem 4: Multilingual_DAPO_0587 (Geometry)

Question. In a right trapezoid $ABCD$ with $AB \parallel CD$ and $\angle B = 75^\circ$, H is the foot of the perpendicular from A to BC. Given $BH = DC$ and $AD + AH = 8$, find AB.
Gold answer: 8. **Solved by:** Korean only (1/10).

Problem 5: Multilingual_DAPO_0771 (Number Theory)

Question. Let $S = \{1, \ldots, 9\}$. Count functions $f : S \to S$ with $f(f(f(s))) = s$ for all s and $3 \nmid (f(s) - s)$ for all s.
Gold answer: 288. **Solved by:** Arabic, Hindi, Swahili (3/10).

Problem 6: Multilingual_DAPO_3014 (Geometry)

Question. A parabola has focus F and directrix ℓ. A focal chord AB has $|AB| = 20$. D and C are projections of A and B onto ℓ, with $|DC| = 14$. Find the area of $ABCD$.
Gold answer: 140. **Solved by:** 8/10 languages. Korean solves in 229 tokens (shortest).

Problem 7: Multilingual_DAPO_0586 (Graph Theory)

Question. At a party, every guest has exactly 14 friends. Every pair of friends shares 6 mutual friends; every pair of non-friends shares 2. How many guests?
Gold answer: 64. **Solved by:** 9/10 languages. Korean solves in 348 tokens (shortest).

Problem 8: Multilingual_DAPO_0882 (Geometry)

Question. Circle with radius $\sqrt{7}$ and center O. Points A, B, C on the circle. $\angle BOC = 120^\circ$ and $AC = AB + 1$. Find AB.
Gold answer: 4. **Solved by:** All 10 languages. Japanese solves in 386 tokens (shortest).

Problem 9: Multilingual_DAPO_0644 (Combinatorics)

Question. How many 3-term geometric sequences a, b, c of positive integers with $a < b < c = 8000$ exist?
Gold answer: 39. **Solved by:** 8/10 languages. Korean solves in 466 tokens (shortest).

Problem 10: Multilingual_DAPO_3582 (Number Theory)

Question. $x_1 = 19, x_2 = 95, x_{n+2} = \text{lcm}(x_{n+1}, x_n) + x_n$. Find $\gcd(x_{2025}, x_{2026})$.
Gold answer: 19. **Solved by:** 9/10 languages. Korean solves in 350 tokens (shortest).

X Future Directions

X.1 Multi-Model Replication

Our analysis uses a single model (DeepSeek-V4-Pro). A critical next step is replicating the key findings across multiple model families:

- **GPT-5.5**: Do proprietary models show the same language-specific reasoning patterns?
- **Qwen-3.5 / Qwen-3.6**: Models with strong Chinese training data may show different Chinese-English dynamics.
- **Gemini 3**: Multilingual by design—do its language-specific effects differ?

X.2 Multilingual Reasoning Ensembles

The most direct practical application is building reasoning ensembles that leverage language diversity:

- **Language selection**: Which subset of languages maximizes the union accuracy? A greedy algorithm starting from English and adding languages by marginal coverage could identify the optimal 3-5 language ensemble.
- **Aggregation strategy**: Beyond majority vote, one could use language-specific confidence weights, verifier models, or answer clustering to combine multilingual solutions.
- **Cost-accuracy tradeoff**: Running 10 languages costs 10× more tokens. Is a 3-language ensemble (e.g., English + Korean + Arabic) sufficient to capture most of the diversity?

X.3 Mechanistic Interpretability

Understanding *why* different languages induce different reasoning paths could yield fundamental insights:

- **Neuron-level analysis**: Do language-specific neurons (Tang et al., 2024) correlate with language-specific mathematical method selection?
- **Attention patterns**: Do attention heads distribute differently across mathematical tokens when the same problem is prompted in different languages?
- **Training data influence**: Can we trace language-specific mathematical instincts to specific training data distributions?

X.4 Beyond Mathematics

Our methodology—prompting in multiple languages and comparing reasoning traces—could be applied to:

- **Logical reasoning**: Do different languages favor different logical frameworks (propositional vs. first-order, constructive vs. classical)?
- **Scientific reasoning**: Does language affect how models approach physics problems, chemistry mechanisms, or biological reasoning?
- **Creative tasks**: Do different languages induce different creative strategies, analogies, or metaphors?

X.5 Evaluation Protocol Design

Our results suggest a new evaluation paradigm:

- **Union-capability reporting**: In addition to per-language accuracy, benchmarks should report the union accuracy—the fraction of problems solved by *any* language.
- **Path diversity metrics**: Benchmarks should measure not just whether the model can solve a problem, but whether it can solve it in multiple distinct ways.
- **Language-as-probe**: Multilingual prompting can be used as a diagnostic tool to understand what mathematical knowledge a model has but cannot access from English alone.