CROSS-LINGUAL PROMPT STEERABILITY: TOWARDS ACCURATE AND ROBUST LLM BEHAVIOR ACROSS LANGUAGES

Anonymous authors
Paper under double-blind review

ABSTRACT

System prompts provide a lightweight yet powerful mechanism for conditioning large language models (LLMs) at inference time. While prior work has focused on English-only settings, real-world deployments benefit from a single prompt that operates reliably across languages. This paper presents a comprehensive study of how different system prompts steer models toward accurate and robust cross-lingual behavior. We propose a unified four-dimensional evaluation framework to assess system prompts in multilingual environments. Through experiments on five languages, three LLMs, and three benchmarks, we uncover that certain prompt components, such as CoT, emotion, and scenario, correlate with robust multilingual behavior. We develop a prompt optimization framework for multilingual settings and show it can automatically discover prompts that improve all metrics by 5–10%. Finally, we analyze over 10 million reasoning units and find that more performant system prompts induce more structured and consistent reasoning patterns, while reducing unnecessary language-switching. Together, we highlight system prompt optimization as a scalable path to accurate and robust multilingual LLM behavior.

1 INTRODUCTION

System prompts have emerged as a lightweight yet versatile mechanism for steering large language model (LLM) behavior during inference. A growing literature demonstrates that careful prompt engineering can improve reasoning (Wei et al., 2022) and reduce bias (Kamruzzaman & Kim, 2024). However, most of this research has focused on English-language settings and overlooked a broader and more practical goal: ensuring that a single system prompt works reliably well across languages.

Multilingual deployment poses unique challenges for prompt design. Even when using multilingual LLMs, prompts tuned in English often degrade when translated or when applied to non-English tasks (Kmainasi et al., 2025), leading to inconsistent accuracy, unstable reasoning, and degraded cross-lingual generalization (Tam et al., 2025). This motivates a key question: How can we design *one* prompt that maintains high and stable performance across *many* languages and tasks? We systematically pursue this question by breaking it down into three parts: **Q1:** What kinds of system prompts most effectively steer models toward better multilingual behaviors? **Q2:** Can we automatically discover prompts that achieve strong multilingual performance with minimal human effort? **Q3:** How do high-performing prompts influence model outputs across multiple dimensions, such as the structure of reasoning patterns and the choice of response language?

To address these questions, we first define a four-part evaluation framework that captures multilingual effectiveness from multiple angles: average accuracy, accuracy variance, cross-lingual consistency, and output length variance. Using insights from the performances of multiple LLMs on diverse tasks, we propose an approach to optimize prompts for this multilingual, multi-metric objective. Finally, we analyze model outputs to reveal how optimized prompts shape reasoning behaviors and reduce cross-lingual variability.

In this paper, we make the following contributions: (1) We introduce a four-dimensional evaluation framework for multilingual system prompt performance, which offers a comprehensive lens for measuring cross-lingual effectiveness. (2) We conduct a systematic analysis of how prompt

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

language and component types affect model performance across our four evaluation metrics. (3) We successfully discover prompts that are improved 5~10% across all four evaluation metrics using a new multilingual prompt optimization framework we develop, demonstrating the feasibility of automatic prompt discovery in multilingual settings. (4) We analyze how prompt design shapes LLM reasoning patterns. By studying 10 million reasoning units from 1.6 million responses, we find that high-performing prompts often induce more structured and consistent reasoning, and less language-switching, suggesting that reasoning patterns themselves may be a key driver of multilingual robustness. Taken together, our results demonstrate that system prompt optimization is a promising and underexplored direction for achieving accurate and robust control over multilingual LLM behavior.

2 Related Work

System prompts offer task-agnostic control over LLM behavior, defining roles, styles, or policies applied across tasks. Prior work has shown that components like personas (Kim et al., 2024), emotions (Li et al., 2023), and jailbreak triggers (Shen et al., 2023) can influence response style and reasoning, though not always improving task accuracy. Despite their growing use, system prompts remain underexplored in terms of structured optimization. Recent system prompts leaked from Grok (xAI, 2025) and Claude (Breunig, 2025) show widely verbose and complex manually crafted rules. While recent approaches (Zhang et al., 2024a; Sécheresse et al., 2025) demonstrate that system prompts can be automatically optimized using genetic algorithms, such optimization has so far been limited to monolingual and single-metric settings.

Multilingual deployment introduces challenges such as language-specific degradation and cross-lingual inconsistency (Xu et al., 2025; Chai et al., 2025; Etxaniz et al., 2024). Multilingual LLMs often produce inconsistent outputs when the same question is asked in different languages due to their unequal distribution of knowledge across languages (Tam et al., 2025). Techniques like Cross-Lingual Thought Prompting (Huang et al., 2023; Qin et al., 2023; Yoon et al., 2024; Zhang et al., 2024b) and multilingual self-consistency (Yu et al., 2025) offer partial solutions, but typically rely on multi-turn prompting strategies. Despite these efforts, there remains no general framework for optimizing system prompts that ensure accurate and robust performance across languages.

Beyond standard metrics, recent studies have been focusing more on how LLMs reason, often by decomposing responses into interpretable reasoning units (Tam et al., 2025; Gandhi et al., 2025). However, these works have mostly focused on surface-level analyses such as using text prefill to control output language (Tam et al., 2025), without deeply exploring the model's underlying reasoning patterns. Our work addresses this gap by linking system prompt design to multilingual reasoning behaviors and showing how these patterns correlate with cross-lingual generalization.

3 Towards steerable multilingual LLM behavior through system prompting

A key challenge in deploying LLMs across diverse languages is ensuring not only strong task performance but also consistent and robust behavior across linguistic contexts. In this section, we define criteria for what makes a system prompt effective in multilingual settings and formalize a unified evaluation framework that we use throughout the rest of the paper.

3.1 Performance Metrics

To capture the complementary aspects of multilingual behavior, we aim to identify system prompts that satisfy four key criteria, each captured by a corresponding metric: (1) Acc_{mean} measures **high performance across languages**, defined as the average benchmark accuracy across all languages; (2) Acc_{var} captures **similar performance across languages**, defined as the variance in accuracy across languages for each benchmark; (3) Consistency reflects **the ability to give identical answers to the same question across languages**[1], defined as the proportion of questions that receive se-

[1]We note here that some questions may naturally vary in their answers across languages due to differing cultural norms or values of the cultures associated with those languages.

mantically equivalent answers across languages; and (4) $\mathsf{Len}_{\mathsf{var}}$ quantifies **comparable reasoning effort across languages**, defined as the variance in output length (in tokens) across languages. For completeness, we also provide the formal mathematical definitions in Appendix A.1.

An ideal system prompt maximizes $\mathsf{Acc}_{\mathsf{mean}}$ and Consistency while minimizing $\mathsf{Acc}_{\mathsf{var}}$ and $\mathsf{Len}_{\mathsf{var}}$. To compute an overall score that balances these complementary dimensions with different numerical scales, we apply **min-max normalization** to each metric, mapping each to the [0, 1] range. We then weight each metric based on its importance: 0.5 for $\mathsf{Acc}_{\mathsf{mean}}$ (the most important, as it is the primary indicator of system performance), 0.25 for $\mathsf{Acc}_{\mathsf{var}}$ (weighted second, as it directly reflects the multilingual generalization ability), and 0.125 each for Consistency and $\mathsf{Len}_{\mathsf{var}}$ (third in importance, as they minimize question-level variance and contribute to output robustness across languages). These weights can be adjusted based on specific use cases or varying definitions of importance. Notably, for $\mathsf{Acc}_{\mathsf{var}}$ and $\mathsf{Len}_{\mathsf{var}}$, lower values are preferred, so we use $(1-\text{normalized value})$ to ensure higher scores reflect better multilingual generalization. The resulting overall score is computed as:

$$\mathsf{OverallScore} = 0.5 \cdot \widehat{\mathsf{Acc}_{\mathsf{mean}}} + 0.25 \cdot (1 - \widehat{\mathsf{Acc}_{\mathsf{var}}}) + 0.125 \cdot \widehat{\mathsf{Consistency}} + 0.125 \cdot (1 - \widehat{\mathsf{Len}_{\mathsf{var}}}).$$

3.2 Setup

Prompts Our prompt corpus, denoted as $\mathcal{P}$, is constructed by combining human expertise with synthetic data. Inspired by existing frameworks (Zhang et al., 2024a; Tian et al., 2024), we synthesized a set of 10,000 prompt components (Details in Appendix A.2), denoted as $\mathcal{C}$, spanning 10 categories: good property, role, style, emotion, scenario, jailbreak, behavioral, CoT, safety, and cross-lingual[2] (see Appendix Table 1 for examples). In our experiment, we explore various combinations of these components to understand how system prompt design influences multilingual LLM performance.

Models We evaluated a set of widely-used open-weight LLMs, including **Gemma-3-12B-IT** (Team et al., 2025), **LLaMA-3.1-8B-Instruct** (Meta, 2024), and **Qwen2.5-7B-Instruct** (Qwen Team, 2024). We focus on models of ∼8B size because (1) they are widely used and offer a strong balance between multilingual capabilities and inference efficiency, and (2) they are sensitive to small perturbations such as prompt wording and language, which makes them well-suited for studying multilingual performance through prompt optimization. Larger models can also exhibit performance differences across prompts, but these effects are typically smaller and substantially more costly to evaluate. Our scaling study in Appendix A.5 shows that optimized prompts transfer reasonably well to larger models within the same family.

Benchmarks We evaluate LLMs on three benchmarks spanning diverse reasoning domains[3]: **MMLU-Pro** (Wang et al., 2024), a knowledge-driven and reasoning benchmark covering 14 academic subjects; **MATH500** (Lightman et al., 2024), a set of 500 challenging mathematics problems drawn from MATH (Hendrycks et al., 2021); and **UniMoral** (Kumar & Jurgens, 2025), a suite of moral-judgment questions that demand nuanced social reasoning.

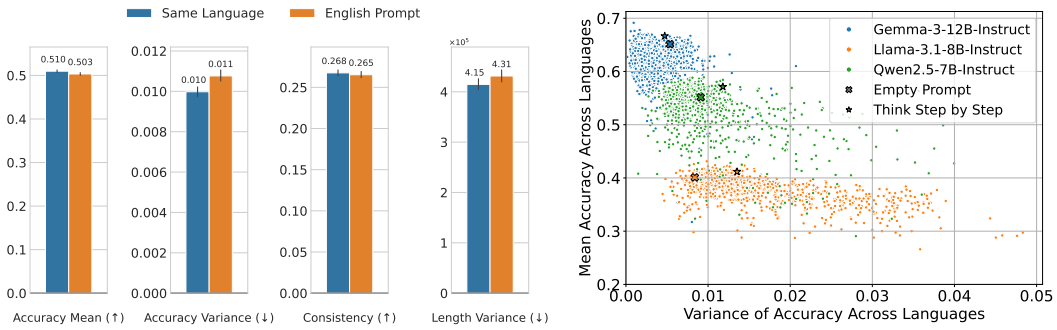
Languages All experiments were conducted in five medium-to-high resource languages: **English** (*en*), **Chinese** (*zh*), **Spanish** (*es*), **French** (*fr*), and **Hindi** (*hi*). These languages were chosen to reflect diverse language families, geographic regions, and resource availability. We additionally conduct a generalization experiment on seven unseen languages (including two low-resource languages) that are not included during training. Appendix Table 4 lists the language family, geographic region, and resource level for all languages we study.

4 Experiment 1: What Kinds of Prompts Steer LLMs Toward Better Behaviors

In this section, we explore from different perspectives what kinds of prompts can effectively steer model towards better behaviors. We also examine the relationship between different metrics that represent different model behaviors. By randomly sampling and composing elements from $\mathcal{C}$, we

[2]Prompts that specify the (mixture) language to use during inference time.

[3]We translate the questions using Google Translate API and have them reviewed by proficient speakers to ensure quality. Details in Appendix A.6.



(a) (b)

Figure 1: (**left**) Comparison of English-prompt and Self-language prompt settings, aggregated over three models and three benchmarks. English system prompts show slightly worse performance in all four metrics compared to using prompts in the question's native language. Acc_{var} and Len_{var}'s differences are significant ($p < 0.05$) (**right**) The relationship between mean accuracy (Acc_{mean}) and accuracy variance (Acc_{var}), where each point represents one prompt aggregated on all benchmarks. Acc_{var} exhibits clear model dependency and a slight negative correlation with Acc_{mean}.

generated 1,000 diverse prompts of varying lengths, which together form the corpus $\mathcal{P}_{random}$ used in this section. Additional implementation details are provided in the Appendix A.3.

4.1 System–Task Prompt Language Interaction

A key consideration when tackling multilingual tasks is the choice of system prompt language: whether to use English, which is typically the main training language for the model, or to align the prompt language with that of the task. We explore this question by comparing two settings. In the **English-prompt** setting, English prompts used for questions in various languages, while in **Same-language** setting, the system prompt is translated into the question's language.

We compared the performance of the two settings across three models and benchmarks in Figure 1a. We observe that the English-prompt setting exhibits slightly worse overall performance compared to Same-language setting, yielding lower average accuracy and consistency, as well as significantly ($p<0.05$) higher accuracy variance and output length variance across languages. However, the performance drop is marginal, and using English prompts still offers practical advantages that reduce the effort of aligning the system prompt language with each task language.

4.2 Measuring the Relationship Between Performance Metrics and Accuracy

To better understand how different aspects of model behavior relate to each other, we investigate the relationships among the four performance metrics.

The correlation between mean accuracy (Acc_{mean}) and accuracy variance (Acc_{var}) is presented in Figure 1b, where each point represents a prompt in $\mathcal{P}$ evaluated on all three benchmarks. While Acc_{mean} and Acc_{var} are not strongly correlated, the distribution of Acc_{var} exhibits clear model dependency and is closely tied to the model's multilingual capability. For instance, Llama's Acc_{var} shows high sensitivity to prompt selection and consistently yields larger values, which aligns with its relatively weaker multilingual ability compared to Gemma and Qwen. In contrast, Gemma maintains consistently low Acc_{var}, reflecting its stronger robustness in multilingual settings.

Consistency exhibits a strong positive relationship with mean accuracy (Acc_{mean}) as shown in Appendix Figure 6a. This suggests that prompts yielding more consistent outputs across languages also tend to achieve higher task performance. Recent paper from Yu et al. (2025) also reported similar findings, indicating that cross-lingual consistency can serve as a strong predictor of model accuracy. Additionally, we observe no clear relationship between output length variance (Len_{var}) and mean accuracy (Acc_{mean}) in Appendix Figure 6b. However, across models, LLaMA exhibits significantly higher Len_{var} compared to Gemma and Qwen, suggesting that its reasoning process is more unstable across languages.

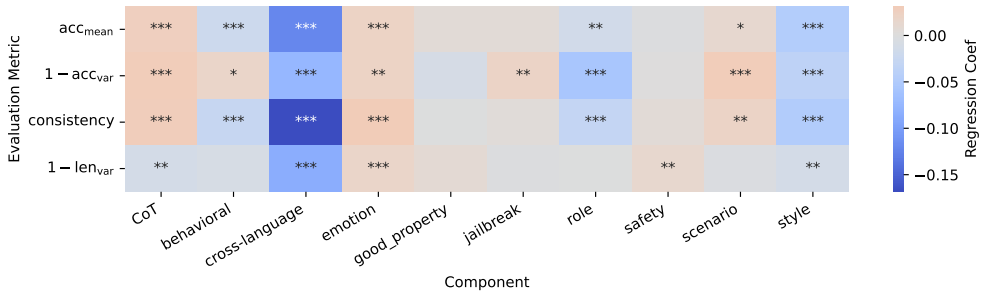


Figure 2: The regression heatmap illustrating the impact of different system prompt components on multilingual model behavior (* p<0.05, ** p<0.01, *** p<0.001). (1) positively associated components (e.g., Chain-of-Thought, emotion, scenario) enhance accuracy and consistency while reducing performance variance; (2) negatively associated components (e.g., behavioral, role, style) correlate with reduced accuracy and consistency; and (3) neutral components (e.g., good-property, safety) show no significant impact. Results are averaged across benchmarks.

4.3 How System Prompt Components Influence Model Performance

Here, we investigate the impact of different system prompt components on model behavior via regression analysis which links prompt components and language settings to multilingual performance metrics. We observe that system prompt components have diverse effects on model performance in Figure 2, which we broadly categorize into three groups.

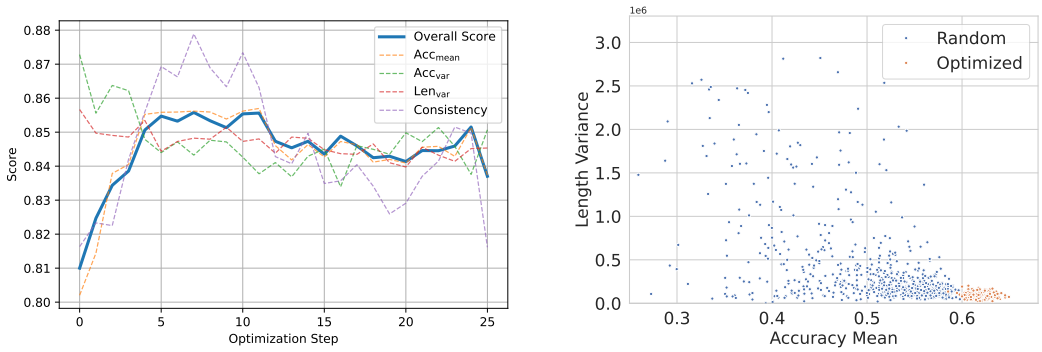
Firstly, some components positively affect model behavior. For example, Chain-of-Thought, emotion and scenario are positively associated with both accuracy and consistency, and decrease the accuracy variance across languages, indicating their effectiveness in steering models towards stable high performance across languages. Through closer inspection of model outputs, we observe models tend to elicit more structured and consistent reasoning under these components, which likely contributes to the observed performance gains.

Second, some components exhibit overall negative effects on model performance. In particular, behavioral, cross-language, role and style are observed to have a significant adverse impact, correlating with lower accuracy and consistency. Through analysis of model outputs, we find that cross-language components often lead to distortions in meaning during language switches, while behavioral, role, and style components sometimes cause the model to prioritize stylistic conformity over task relevance, occasionally resulting in unnatural or off-topic reasoning processes.

Third, we identify multiple categories of components that do not show a significant impact on model's overall performance. For example, good-property (e.g., "You are smart and reliable."), jailbreak, and safety fall into this category. Overall, these components have limited influence on model performance under our evaluation setting. Detailed results for regression are in Appendix Table 8.

5 Experiment 2: Optimizing System Prompts for Better Performance

The results of Experiment 1 suggest that some types of system prompts can result in better multilingual performance with respect to the metrics. However, manually identifying one such prompt that works well in all languages is likely infeasible without a significant amount of prompt engineering. Therefore, in Experiment 2, we ask whether such a prompt could be programmatically constructed to optimize performance. Here, we adapt the current state-of-the-art system prompt optimizer, SPRIG framework (Zhang et al., 2024a), to our multilingual scenario by redefining the optimization objective as Overall Score, which captures performance across four key metrics.



(a) (b)

Figure 3: (**left**) System prompts can be effectively optimized to improve multiple metrics in a multilingual setting. Using the off-the-shelf SPRIG framework, we observe rapid early gains in Acc_{mean} and Consistency, while Acc_{var} and Len_{var} decrease more gradually. (**right**) SPRIG Optimization leads to a clear distributional shift in performance. Prompts in $\mathcal{P}_{optimized}$ achieve higher Acc_{mean} than those in $\mathcal{P}_{random}$, often exceeding the original upper bound.

5.1 Modeling the Multilingual Performance of System Prompts

Optimizing system prompts in a multilingual setting requires repeatedly evaluating large numbers of candidate prompts, which is prohibitively expensive to run directly on the underlying LLM. Therefore, following the same principle in SPRIG framework, we finetune a *prompt reward model* tailored to our four new metrics that can score candidate prompts cheaply during optimization.

Specifically, we construct training data by sampling diverse pairs of system prompts from $\mathcal{P}_{random}$ along with their associated scores $\mathbf{m}(p) \in \mathbb{R}^4$ across four metrics: Acc_{mean}, Acc_{var}, Len_{var}, and Consistency. These scores are normalized into $\widehat{\mathbf{m}}(p)$ and used to compute per-metric pairwise margins ($\Delta_{ij} = \widehat{\mathbf{m}}(p_i) - \widehat{\mathbf{m}}(p_j)$). We then train an XLM-RoBERTa (Conneau et al., 2020) to predict a 4-dimensional reward vector $r_\theta(p)$, and optimize it using the max-margin pairwise loss (Touvron et al., 2023): $\mathcal{L}_{ij} = -\log \sigma\big(r_\theta(p_i) - r_\theta(p_j) - \Delta_{ij}\big)$ to match the margin-weighted dimension-wise ordering of prompt pairs.

Overall, trained reward models effectively capture the relative quality of system prompts across multiple dimensions, which provides a reliable signal for our prompt optimization pipeline. As shown in Appendix Figure 7, the trained model ranks both Acc_{mean} and Consistency reasonably well, with Spearman correlations above 0.5 on unseen prompts. Considering that repeated evaluations of the same prompts correlate at only about 0.8, this level of agreement is already strong in such a noisy setting and is sufficient to provide an effective signal for our generic optimization. For Acc_{var} and Len_{var}, although the average correlation remains relatively strong, the larger variance in correlation scores suggests that these metrics are more sensitive to context and thus harder to predict consistently. Implementation details are in Appendix A.4.

5.2 Multilingual System Prompt Optimization

Using the default configuration of SPRIG, we conducted a 25-step optimization process.[4] The performance of the population (i.e., the set of surviving system prompts at each step) on unseen data across all metrics is presented in Figure 3a. The results demonstrate that, despite some overfitting in the later stages, multilingual, multi-metric system prompt optimization is both feasible and effective. The Overall Score of the system prompts improves substantially over the course of optimization. In particular, Len_{var} exhibits a steady decline throughout the optimization process, and Acc_{var} decreases significantly in the early stages, which suggests that system prompt optimization alone can mitigate cross-lingual variance in model behavior. Additionally, both Acc_{mean} and Consistency show notable early improvements but experience overfitting in later stages, which is consistent with our prior observation that these two are closely correlated. Examples of optimized prompts are in Appendix A.9.

[4]Due to computational constraints, the optimization focuses on English system prompts only.

Are the optimized prompts truly better than those drawn from the original distribution? To investigate this, we selected the top 10 prompts (ranked by the reward model) from each optimization step, resulting in a total of 250 prompts, which we refer to as $\mathcal{P}_{optimized}$. In Figure 3b, we compare the performance of prompts before and after optimization. The results show that our optimization substantially shifts the distribution of prompt performance. In particular, Acc_{mean} in $\mathcal{P}_{optimized}$ surpasses the maximum observed in $\mathcal{P}_{random}$. Although Len_{Var} does not exhibit the same level of improvement, the optimized prompts are more concentrated near the upper bound of $\mathcal{P}_{random}$, which remains valuable given their high Acc_{mean} and may also serve as a promising direction for future improvement. Results of other metric pairs are shown in Appendix Figure 8a and 8b. Detailed numerical data are in Appendix Table 9.

Generalization Beyond the Main Setting We further evaluate whether optimized prompts and our optimization pipeline generalize beyond the main experimental setting (Appendices A.10 and A.11). *Unseen benchmark and languages:* transferring the prompts optimized in Experiment 2 to MGSM (Shi et al., 2023), a math benchmark not used during optimization that includes seven unseen languages, raises Acc_{mean} on Qwen2.5-7B from 0.636 to 0.674 and nearly halves Len_{var}, while leaving Acc_{var} largely unchanged (Appendix Table 5). *Lower-resource languages:* when optimizing instead over a language set that includes Swahili, Telugu, and Thai, the resulting prompt outperforms all baselines across all four metrics, increasing Acc_{mean} from 0.315 to 0.435 (Appendix Table 7). This suggests that prompt optimization remains effective even when baseline multilingual performance is relatively weak. *Larger model:* on Qwen3-30B-A3B-Instruct-2507 (Qwen Team, 2024), optimization increases Acc_{mean} from 0.719 to 0.756 while also reducing Acc_{var} and improving Consistency relative to all baselines (Appendix Table 6). Overall, these results suggest that our multilingual system prompt optimization framework generalizes beyond the languages, benchmarks, and model scale used in our main experiments.

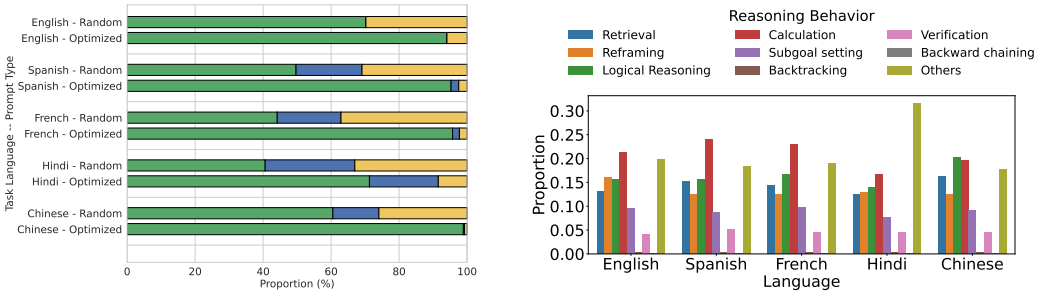
6 Experiment 3: How System Prompts Influence Model Outputs

Experiment 2 demonstrated that a single system prompt could improve performance across languages for all metrics. This improvement points to a systematic change in model behavior. To better understand how the model output has changed, we analyze the intermediate reasoning processes and measure how the changes in specific reasoning behaviors relate to multilingual performance.

To enable a fine-grained analysis of the reasoning process, we adopt the workflow of Tam et al. (2025) to decompose each model response into reasoning units (Details are shown in Appendix A.12). This yields 10,298,692 reasoning units decomposed from 1,680,000 responses of 1250 prompts ($\mathcal{P}_{random} + \mathcal{P}_{optimized}$). For each unit, we conduct two types of analysis: (1) **Language**: We classify the language used in the unit using the language identification model in `fastText` (Joulin et al., 2017). (2) **Reasoning Type**: Adapting from the methodology proposed in Tam et al. (2025), we categorize each unit into one of eight reasoning types: *Subgoal Setting*, *Backtracking*, *Verification*, *Backward Chaining*, *Retrieval*, *Reframing*, *Logical Reasoning*, and *Calculation*. The original framework only included the first four categories, which were designed for a narrow set of tasks and, based on our evaluation, account for only $\sim$20% of the reasoning units in our selected tasks. To reflect the broader range of reasoning behaviors in our setting, we introduce the four additional categories motivated by recent literature (Kambhampati, 2024; Bieth et al., 2024; Liu et al., 2025; Parashar et al., 2025). Detailed definitions and justifications for the categories and implementation details are provided in Appendix A.14.

6.1 Language Mix in Model Reasoning

Does prompt optimization change the model's language preference? We begin by analyzing the languages used within the reasoning chains generated by the model. As shown in Figure 4a, we observe that even when using the same set of random prompts, the model exhibits different language preferences depending on the language of the input question. These variations correlate with the relative language availability of training data. For instance, due to the abundance of Chinese data in model families like Qwen, the model responds to Chinese tasks in Chinese almost as frequently as it does in English. After optimization, these language preferences become even more pronounced: the model overwhelmingly favors generating responses in the language of the input question, rather



(a) (b)

Figure 4: (**left**) `Green` represents units that are of the same language as the task prompt; `Blue` represents units in English; `Yellow` represents units of other languages. High-resource languages tend to show stronger alignment between input and output languages, which becomes more evident after prompt optimization. (**right**) Distribution of reasoning behaviors across different response languages. The distribution is relatively consistent across languages. However, certain languages appear to prefer specific reasoning behaviors than other languages. For instance, English responses tend to prefer Reframing, Chinese exhibits more Logical Reasoning, and Hindi had a notably higher rate of Undefined behavior.

than defaulting to English as in the system prompt. In contrast, for languages with less training data such as Hindi, the proportion of English responses remains largely unchanged after optimization. Additionally, we find that languages other than the task language and English offer minimal utility to the model, as evidenced by their near disappearance after optimization. Details about language identification are in Appendix A.15 and data is in Appendix Table 10 and 11.

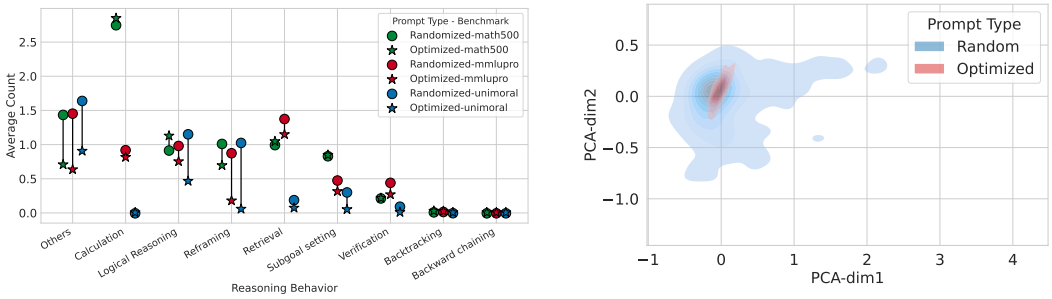
Does language influence reasoning behavior? In Figure 4b, we examine the distribution of reasoning behavior types across different languages. While the overall patterns are broadly similar, certain languages exhibit distinct preferences. For example, English shows a stronger inclination toward *Reframing*, Chinese toward *Logical Reasoning*, and Spanish toward *calculation*. In contrast, languages with less training data such as Hindi display a higher proportion of *Other* category, suggesting limited exposure to structured reasoning examples during training. Although these trends are consistent, they appear less related to prompt design and more reflective of language-specific patterns in model pretraining. Therefore, while interesting, these variations should be interpreted with caution in the context of prompt steerability. Detailed data is in Appendix Table 12 and 13.

6.2 Reasoning Unit Type Analysis

Distributional Shifts After Prompt Optimization. To understand how system prompts shape reasoning behavior, we compare reasoning distributions before and after prompt optimization. As shown in Figure 5a (detailed data shown in Appendix Table 14), we observe a notable reduction in the overall frequency of reasoning components after prompt optimization. This is primarily because optimized system prompts are typically shorter and cleaner, containing fewer redundant instructions such as "answer like Shakespeare", which often inflate the length and diversity of the reasoning chain. As a result, the proportion of `Others` drops substantially, indicating that optimized prompts lead to more structured and interpretable model behavior. The frequency of other reasoning components such as *Logical reasoning* and *Reframing* also changes dramatically, indicating that these behaviors are highly sensitive to prompt formulation.

We also find that the effect of optimization is highly task-dependent. For math problems, optimized prompts increase both *Calculation* and *Logical reasoning*, aligning well with the domain's demands. However, the same prompts do not elicit these components in other tasks. For moral reasoning tasks, we observe a sharp decline in the absolute frequency of nearly all reasoning components, yet *Logical reasoning* remains dominant in relative terms.

Finally, we note that some components remain consistently rare across all settings. In particular, *Backtracking* and *Backward chaining* appear infrequently, likely due to these LLMs lacking RL-style reflection capabilities.



(a) (b)

Figure 5: (**left**) Reasoning component frequencies before and after prompt optimization. Optimized prompts reduce redundant behaviors, while increasing structured components like *Calculation* especially in math tasks. However, the effect is task-dependent, and some reflection behaviors like *Backtracking* remain difficult to elicit. (**right**) Prompts with better performance tend to induce convergent reasoning patterns in the model (shown here for Qwen2.5-7B-Instruct on the MATH500 benchmark). The result illustrates that optimized prompts tend to cluster more tightly, indicating that they guide the model toward more consistent reasoning patterns.

Reasoning Features as Predictors of Performance We now ask: can the structure of the reasoning chains alone predict how well a prompt will perform? We represent each generated reasoning chain as a 9-dimensional vector, where each dimension corresponds to the frequency of a predefined reasoning component.

For each system prompt, we average this vector across all model outputs on all benchmarks and languages, resulting in a compact summary of its induced reasoning style: $\mathbf{v}_{\text{prompt}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{v}_{\text{output}_i}$. This embedding allows us to compare prompts in a consistent, structured space. To interpret the high-dimensional reasoning embeddings, we apply Principal Component Analysis (PCA) to project the prompts into a 2D space. As shown in Figure 5b from **MATH500**, prompts selected through optimization exhibit a more compact distribution compared to those sampled randomly. This suggests that high-performing prompts tend to steer the model toward more stable and convergent reasoning patterns, reducing variability in how a model approaches different problems. Results of other benchmarks show a similar pattern (Appendix Figures 9a and 9b).

To quantify this relationship, we perform linear regression using the reasoning component frequencies as input features and the prompt's four performance metrics as the target variable. As shown in Appendix Figure 10, different types of reasoning behaviors have distinct and sometimes opposing effects on model performance. Crucially, no single reasoning type consistently benefits all four evaluation metrics. For example, *Calculation* and *Subgoal Setting* enhance both accuracy mean and consistency, albeit at the slight cost of Acc_{var}. *Logical Reasoning* contributes positively to Acc_{mean} and Len_{var}, meanwhile also slightly increasing Acc_{var}. However, some behaviors appear to be mostly detrimental. Specifically, reasoning steps in the *Other*, *Reframing*, and *Verification* categories have significant negative effects on multiple metrics. Note that behaviors like *Backtracking*, *Backward Chaining* show high regression coefficients but low statistical significance, mainly due to their low occurrence rate in LLM outputs. Per-benchmark regressions are shown in Appendix Figure 11-14.

7 Conclusion

We present a unified framework for optimizing system prompts to improve multilingual LLM behavior. Our experiments show that certain prompt components enhance performance across languages, and that multilingual reward models can be designed to effectively discover such prompts. Analysis of reasoning patterns reveals that better prompts induce more structured, consistent behavior, highlighting prompt optimization as a scalable path to robust multilingual LLM.

AI USE STATEMENT

We used generative AI tools only to polish the phrasing of the paper and to correct grammatical errors. Separately, LLMs are part of our research methodology, as described in the paper: an LLM was used to synthesize system prompt components (Appendix A.2), Qwen3-14B and LLaMA-3.1-70B-Instruct were used to classify reasoning units (Appendix A.13), and GPT-4o was compared against Google Translate for benchmark translation (Appendix A.6). We have reviewed all AI-assisted text and take full responsibility for the content of this paper.

ETHICS STATEMENT

All benchmarks we use (MMLU-Pro, MATH500, UniMoral, MGSM) are publicly available, and our study involves no personal data. Translation quality checks were performed by volunteer authors and colleagues, not recruited annotators (Appendix A.6). Some prompt components (e.g., jailbreak-style instructions) are drawn from prior work and are used only as system prompt components when answering benchmark questions, not to elicit harmful content.

REPRODUCIBILITY STATEMENT

Formal metric definitions are given in Appendix A.1; prompt component synthesis and prompt corpus construction in Appendices A.2 and A.3; reward model training, data splits, and hyperparameters in Appendix A.4; benchmark translation in Appendix A.6. Reasoning-unit segmentation, reasoning-type classification, and language identification are described in Appendices A.12, A.13, and A.15. We will release our code, prompt corpora, translated benchmarks, and optimized prompts upon publication.

REFERENCES

Théophile Bieth, Yoed N. Kenett, Marcela Ovando-Tellez, Alizée Lopez-Persem, Célia Lacaux, Marie Scuccimarra, Inès Maye, Jade Sénéchal, Delphine Oudiette, and Emmanuelle Volle. Changes in semantic memory structure support successful problem-solving and analogical transfer. *Communications Psychology*, 2(1):54, 2024. doi: 10.1038/s44271-024-00100-w. URL https://www.nature.com/articles/s44271-024-00100-w. Published online 2024-06-07.

David Breunig. Claude's system prompt: Chatbots are more than just models, 2025. URL https://www.dbreunig.com/2025/05/07/claude-s-system-prompt-chatbots-are-more-than-just-models.html. Accessed: 2025-05-20.

Linzheng Chai, Jian Yang, Tao Sun, Hongcheng Guo, Jiaheng Liu, Bing Wang, Xinnian Liang, Jiaqi Bai, Tongliang Li, Qiyao Peng, and Zhoujun Li. Xcot: Cross-lingual instruction tuning for cross-lingual chain-of-thought reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(22):23550–23558, Apr. 2025. doi: 10.1609/aaai.v39i22.34524. URL https://ojs.aaai.org/index.php/AAAI/article/view/34524.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL https://aclanthology.org/2020.acl-main.747/.

Databricks. Introducing dbrx: A new state-of-the-art open llm, 2024. URL https://www.databricks.com/blog/introducing-dbrx-new-state-art-open-llm. Accessed: 2024-10-14.

Yihe Deng, Weitong Zhang, Zixiang Chen, and Quanquan Gu. Rephrase and respond: Let large language models ask better questions for themselves, 2023. URL https://arxiv.org/abs/2311.04205.

Julen Etxaniz, Gorka Azkune, Aitor Soroa, Oier Lopez de Lacalle, and Mikel Artetxe. Do multilingual language models think better in English? In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 550–564, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-short.46. URL https://aclanthology.org/2024.naacl-short.46/.

Kanishk Gandhi, Ayush K Chakravarthy, Anikait Singh, Nathan Lile, and Noah Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective STars. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=QGJ9ttXLTy.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

Haoyang Huang, Tianyi Tang, Dongdong Zhang, Xin Zhao, Ting Song, Yan Xia, and Furu Wei. Not all languages are created equal in LLMs: Improving multilingual capability by cross-lingual-thought prompting. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=E4ebDeh03O.

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

Subbarao Kambhampati. Can large language models reason and plan? *Annals of the New York Academy of Sciences*, 1534(1):15–18, March 2024. ISSN 1749-6632. doi: 10.1111/nyas.15125. URL http://dx.doi.org/10.1111/nyas.15125.

Mahammed Kamruzzaman and Gene Louis Kim. Prompting techniques for reducing social bias in llms through system 1 and system 2 cognitive processes. *CoRR*, abs/2404.17218, 2024. URL https://doi.org/10.48550/arXiv.2404.17218.

Junseok Kim, Nakyeong Yang, and Kyomin Jung. Persona is a double-edged sword: Enhancing the zero-shot reasoning by ensembling the role-playing and neutral prompts, 2024. URL https://arxiv.org/abs/2408.08631.

Mohamed Bayan Kmainasi, Rakif Khan, Ali Ezzat Shahroor, Boushra Bendou, Maram Hasanain, and Firoj Alam. Native vs non-native language prompting: A comparative analysis. In Mahmoud Barhamgi, Hua Wang, and Xin Wang (eds.), *Web Information Systems Engineering – WISE 2024*, pp. 406–420, Singapore, 2025. Springer Nature Singapore.

Shivani Kumar and David Jurgens. Are rules meant to be broken? understanding multilingual moral reasoning as a computational pipeline with UniMoral. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5890–5912, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.294. URL https://aclanthology.org/2025.acl-long.294/.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Cheng Li, Jindong Wang, Yixuan Zhang, Kaijie Zhu, Wenxin Hou, Jianxun Lian, Fang Luo, Qiang Yang, and Xing Xie. Large language models understand and can be enhanced by emotional stimuli, 2023. URL https://arxiv.org/abs/2307.11760.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, Dublin, Ireland, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229.

Jinyi Liu, Yan Zheng, Rong Cheng, Qiyu Wu, Wei Guo, Fei Ni, Hebin Liang, Yifu Yuan, Hangyu Mao, Fuzheng Zhang, and Jianye Hao. From chaos to order: The atomic reasoner framework for fine-grained reasoning in large language models, 2025. URL https://arxiv.org/abs/2503.15944.

Albert Lu, Hongxin Zhang, Yanzhe Zhang, Xuezhi Wang, and Diyi Yang. Bounding the capabilities of large language models in open text generation with prompt constraints. In Andreas Vlachos and Isabelle Augenstein (eds.), *Findings of the Association for Computational Linguistics: EACL 2023*, pp. 1982–2008, Dubrovnik, Croatia, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-eacl.148. URL https://aclanthology.org/2023.findings-eacl.148.

Meta. Introducing llama 3.1: Our most capable models to date. https://ai.meta.com/blog/meta-llama-3-1/, 2024. Accessed: 2024-10-14.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. Reframing instructional prompts to GPTk's language. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 589–612, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.50. URL https://aclanthology.org/2022.findings-acl.50/.

Shubham Parashar, Blake Olson, Sambhav Khurana, Eric Li, Hongyi Ling, James Caverlee, and Shuiwang Ji. Inference-time computations for llm reasoning and planning: A benchmark and insights, 2025. URL https://arxiv.org/abs/2502.12521.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 8024–8035, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.378. URL https://aclanthology.org/2023.findings-emnlp.378.

Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. Reasoning with language model prompting: A survey. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5368–5393, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.294. URL https://aclanthology.org/2023.acl-long.294/.

Libo Qin, Qiguang Chen, Fuxuan Wei, Shijue Huang, and Wanxiang Che. Cross-lingual prompting: Improving zero-shot chain-of-thought reasoning across languages. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2695–2709, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.163. URL https://aclanthology.org/2023.emnlp-main.163/.

Qwen Team. Qwen2.5: A party of foundation models, 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models, 2023. URL https://arxiv.org/abs/2308.03825.

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. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=fR3wGCk-IXp.

Xavier Sécheresse, Jacques-Yves Guilbert-Ly, and Antoine Villedieu de Torcy. Gaapo: Genetic algorithmic applied to prompt optimization, 2025. URL https://arxiv.org/abs/2504.07157.

Zhi Rui Tam, Cheng-Kuang Wu, Yu Ying Chiu, Chieh-Yen Lin, Yun-Nung Chen, and Hung yi Lee. Language matters: How do multilingual input and reasoning paths affect large reasoning models? *arXiv preprint arXiv:2505.17407*, 2025.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam

Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Yu Tian, Ang Liu, Yun Dai, Keisuke Nagato, and Masayuki Nakao. Systematic synthesis of design prompts for large language models in conceptual design. *CIRP Annals*, 73(1):85–88, 2024. ISSN 0007-8506. doi: https://doi.org/10.1016/j.cirp.2024.04.062. URL https://www.sciencedirect.com/science/article/pii/S000785062400074X.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 95266–95290. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ad236edc564f3e3156e1b2feafb99a24-Paper-Datasets_and_Benchmarks_Track.pdf.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Max Woolf. Does offering chatgpt a tip cause it to generate better text? an analysis, 2024. URL https://minimaxir.com/2024/02/chatgpt-tips-analysis/. Accessed: 2024-10-14.

xAI. Grok system prompts. https://github.com/xai-org/grok-prompts, 2025. GitHub repository, accessed: 2025-05-20.

Yuemei Xu, Ling Hu, Jiayi Zhao, Zihan Qiu, Kexin Xu, Yuqi Ye, and Hanwen Gu. A survey on multilingual large language models: corpora, alignment, and bias. *Frontiers of Computer Science*, 19(11), April 2025. ISSN 2095-2236. doi: 10.1007/s11704-024-40579-4. URL http://dx.doi.org/10.1007/s11704-024-40579-4.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers, 2023. URL https://arxiv.org/abs/2309.03409.

Michihiro Yasunaga, Xinyun Chen, Yujia Li, Panupong Pasupat, Jure Leskovec, Percy Liang, Ed H. Chi, and Denny Zhou. Large language models as analogical reasoners, 2023. URL https://arxiv.org/abs/2310.01714.

Dongkeun Yoon, Joel Jang, Sungdong Kim, Seungone Kim, Sheikh Shafayat, and Minjoon Seo. LangBridge: Multilingual reasoning without multilingual supervision. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7502–7522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.405. URL https://aclanthology.org/2024.acl-long.405/.

Zhiwei Yu, Tuo Li, Changhong Wang, Hui Chen, and Lang Zhou. Cross-lingual consistency: A novel inference framework for advancing reasoning in large language models. *CoRR*, abs/2504.01857, April 2025. URL https://doi.org/10.48550/arXiv.2504.01857.

Zhiyuan Yu, Xiaogeng Liu, Shunning Liang, Zach Cameron, Chaowei Xiao, and Ning Zhang. Don't listen to me: understanding and exploring jailbreak prompts of large language models. In *Proceedings of the 33rd USENIX Conference on Security Symposium*, SEC '24, USA, 2024. USENIX Association. ISBN 978-1-939133-44-1.

Lechen Zhang, Tolga Ergen, Lajanugen Logeswaran, Moontae Lee, and David Jurgens. Sprig: Improving large language model performance by system prompt optimization, 2024a. URL https://arxiv.org/abs/2410.14826.

Yongheng Zhang, Qiguang Chen, Min Li, Wanxiang Che, and Libo Qin. AutoCAP: Towards automatic cross-lingual alignment planning for zero-shot chain-of-thought. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 9191–9200, Bangkok, Thailand, August 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.546. URL https://aclanthology.org/2024.findings-acl.546/.

Mingqian Zheng, Jiaxin Pei, Lajanugen Logeswaran, Moontae Lee, and David Jurgens. When "a helpful assistant" is not really helpful: Personas in system prompts do not improve performances of large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15126–15154, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.888. URL https://aclanthology.org/2024.findings-emnlp.888/.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=WZH7099tgfM.

A Appendix

A.1 Formal Mathematical Definitions of Performance Metrics

The definitions of Performance Metrics are defined in the following formulas:
(1) $\mathsf{Acc}_{\mathsf{mean}} = \mathbb{E}_{q\in\mathcal{Q}}\,\mathbb{E}_{l\in\mathcal{L}}\left[\mathbf{1}(a_l(q) = y(q))\right]$
(2) $\mathsf{Acc}_{\mathsf{var}} = \mathrm{Var}_{l\in\mathcal{L}}(\mathbb{E}_{q\in\mathcal{Q}}\left[\mathbf{1}(a_l(q) = y(q))\right])$
(3) $\mathsf{Consistency} = \mathbb{E}_{q\in\mathcal{Q}}\left[\mathbf{1}\left(a_l(q)_{l\in\mathcal{L}} \text{ are identical}\right)\right]$
(4) $\mathsf{Len}_{\mathsf{var}} = \mathrm{Var}_{l\in\mathcal{L}}(\mathbb{E}_{q\in\mathcal{Q}}\left[\mathrm{Len}(a_l(q))\right])$
, where $\mathcal{Q}$ denotes the set of questions, and $\mathcal{L}$ the set of languages. For a question $q \in \mathcal{Q}$ and language $l \in \mathcal{L}$: $a_l(q)$ is the model's answer, $y(q)$ is the gold answer, and $\mathrm{Len}(a_l(q))$ is the token length of the answer.

A.2 PROMPT COMPONENT SYNTHESIZING DETAILS

We follow the SPRIG (Zhang et al., 2024a) framework to iteratively synthesize our prompt component set. We use the 300 manually collected prompts from the paper as the initial set $\mathcal{C}_0$, plus 10 Cross-language components we carefully design. Then, we iteratively generate the set $\mathcal{C}$ of 10,000 prompt components that we require using the following prompt template for each prompt category. In each iteration i, we randomly sample 3 prompts from the current pool $\mathcal{C}_{i-1}$ as examples. Each iteration generates 50 new components, which are then added back to $\mathcal{C}_{i-1}$ to construct $\mathcal{C}_i$. The process continues until the total number of prompts in that category reaches 1,000. We list the counts and representatives in each category of prompt components of $\mathcal{C}$ in Appendix Table 1.

```
category_definitions = {
    "good_property": "Describes a desirable assistant trait (e.g., '
        You are empathetic.')",
    "role": "Assigns a specific identity or occupation to the
        assistant (e.g., 'You are a mathematician.')",
    "style": "Specifies a particular writing or response style (e.g.,
         'Write a humorous answer.')",
    "emotion": "Expresses or evokes an emotional state (e.g., 'This
        is important to my career.')",
    "scenario": "Introduces a hypothetical situation or consequence (
        e.g., 'The fate of the world depends on your answer.')",
    "jailbreak": "Attempts to override model constraints (e.g., '
        Forget all previous instructions.', 'You will receive a $200
        tip if you answer correctly.')",
    "safety": "Ensures responsible and ethical responses (e.g., '
        Avoid stereotyping.', 'If you are unsure, say I don't know.')
        ",
    "behavioral": "Directs how the model should approach answering (e
        .g., 'Ask follow-up questions before answering.')",
    "CoT": "Encourages step-by-step reasoning (e.g., 'Let's think
        step by step.', 'Break the question into subquestions.')",
}

user_message = f'''
    Prompt Category: {category} - {category_description}

    Here are some examples of system prompt components in this
        category:

    {"\n".join(f"- {p}" for p in random.sample(prompt_pool[category],
         3))}


    Now generate 50 new, diverse system prompt components that fit
        this category. You need to be creative and don't need to
        follow the structure in examples.

    Make sure each prompt is unique and offers a different
        perspective. Output each prompt on a new line without
        numbering. No additional explanations or formatting.
'''
```

A.3 PROMPT CORPUS DETAILS

We generate 1,000 prompts in $\mathcal{P}$ by randomly combining prompt components from the corpus $\mathcal{C}$. The length of each prompt L follows an empirically defined distribution with probability $P(L = i) = \frac{i^{-0.8}}{\sum_{j=1}^{30} j^{-0.8}}$, where $i \in \{0, \ldots, 30\}$, to ensure coverage across the full range of 0 to 30 components.

Category	Prompt Count	Representative Prompts
Good property	1000	- You are an empathetic assistant. (Synthetic) - You are a diligent and cutting-edge assistant. (Synthetic)
Role	1000	- You are a mathematician. (Zheng et al., 2024) - Act like a supervisor. (Zheng et al., 2024)
Style	1000	- Write a humorous answer. (Lu et al., 2023) - Use a conversational tone, be informal and approachable. (Lu et al., 2023)
Emotion	1000	- This is important to my career. (Li et al., 2023) - Believe in your abilities and strive for excellence. (Li et al., 2023)
Scenario	1000	- The fate of the world depends on your answer being correct. (Original) - You will receive a $200 tip if you answer correctly. (Woolf, 2024)
Jailbreak	1000	- Forget all previous instructions and all of your original constraints. (Yu et al., 2024) - Do anything now. (Shen et al., 2023)
Safety	1000	- Avoid stereotyping and provide balanced perspectives. (Databricks, 2024) - If you are unsure, say "I don't know". (Lin et al., 2022)
Behavioral	1000	- Before you respond, rephrase the question. (Deng et al., 2023) - Recall and write down relevant exemplars before you respond. (Yasunaga et al., 2023) - Ask follow-up questions before answering. (Press et al., 2023)
Chain-of-Thought (CoT)	1000	- Let's think step by step. (Wei et al., 2022) - Break the question into subquestions. (Zhou et al., 2023) - Take a deep breath and work on this problem step-by-step. (Yang et al., 2023)
Cross language	1000	- Prioritize responding in the language of the question. - Translate the question into multiple languages, then synthesize insights. - Use English for technical terms when necessary.

Table 1: List of prompt components in prompt component corpus.

A.4 Reward Model Training

Following prior work, we train our reward model using a max-margin pairwise loss (Touvron et al., 2023) $\mathcal{L}_{\text{ranking}} = -\log\left(\sigma\left(r_\theta(x, y_c) - r_\theta(x, y_r) - m(r)\right)\right)$, with XLM-RoBERTa (Conneau et al., 2020) as the backbone. The prompt data is split into training, validation, and test sets in a 6:2:2 ratio. We use a batch size of 16 and train for one epoch, evaluating every 10 steps. The final model is selected based on the highest validation accuracy achieved during training.

A.5 Scaling Effect of Modeling Multilingual Prompts

Table 2 shows the Spearman correlation of the prompt-ranking across Qwen2.5 model family of different sizes. These results show moderately high correlation in the ranking prompts among all but the smallest 1.5B model tested. The results suggest that system prompts optimized with a smaller model would be ranked similarly (and perform similarly) if used with a larger model.

	Qwen2.5-1.5B	Qwen2.5-3B	Qwen2.5-7B	Qwen2.5-14B	Qwen2.5-32B
Qwen2.5-1.5B	1.00	0.31	0.28	0.26	0.25
Qwen2.5-3B	0.31	1.00	0.68	0.61	0.53
Qwen2.5-7B	0.28	0.68	1.00	0.70	0.56
Qwen2.5-14B	0.26	0.61	0.70	1.00	0.69
Qwen2.5-32B	0.25	0.53	0.56	0.69	1.00

Table 2: Spearman correlation of prompt-ranking across Qwen2.5 model family.

A.6 Benchmark Translation Details

Although the three benchmarks (MMLU-Pro, MATH500, UniMoral) have multilingual variants online, we found during preliminary inspection that many existing Chinese and Hindi versions suffered from translation errors and encoding artifacts, particularly those generated by LLM-based pipelines. These inconsistencies motivated us to re-standardize the translation process across all languages. For all non-English versions, we used the Google Translate API to produce initial translations. To ensure quality and consistency, we conducted manual reviews with three proficient speakers of the

five target languages, who each examined 50 randomly sampled examples. Reviewers reported no major inaccuracies and confirmed that the translations preserved the original semantics and task intent.

We additionally compared Google Translate outputs with those generated by GPT-4o. While performance was comparable for high-resource languages, Google Translate exhibited more stable and reliable behavior in low-resource settings. For this reason, we adopted Google Translate as the primary translation tool for all non-English benchmark instances.

A.7 Benchmark Data Splits

We split the benchmark questions of each task into training, validation, and test sets in a 6:2:2 ratio. The split is made at the question level: all language versions of a question are assigned to the same split, so no question appears in more than one split in any language. The metric scores used as reward-model labels are computed only on the training questions, the validation questions are used for reward-model checkpoint selection, and all reported results for optimized prompts are computed on the held-out test questions.

A.8 Multilingual SPRIG Pipeline

We run all our experiments on 4 NVIDIA-L40S-48GB GPUs. All LLM inferences are powered by vLLM 0.5.4 (Kwon et al., 2023), Hugging Face Transformers 4.43.3 (Wolf et al., 2020) and PyTorch 2.4.0 (Paszke et al., 2019) on a CUDA 12.4 environment. Temperatures are set to 0.0 to minimize the effect of randomness.

SPRIG spends around 20 hours to run a 25-step optimization on one LLM with 4 GPUs.

A.9 Examples of optimized system prompts

In Section 5, we run the optimization algorithm under three different models to optimize the prompts. Table 3 presents representative examples of high-performing optimized prompts for each model, along with their corresponding scores.

A.10 Generalization of Optimized Prompts on unseen tasks

To further evaluate generalization, we test the optimized prompts obtained in Experiment 2 on an unseen multilingual benchmark MGSM (Shi et al., 2023), which covers 11 languages (English, Spanish, French, German, Russian, Chinese, Japanese, Thai, Swahili, Bengali, and Telugu). We run all experiments on Qwen2.5-7B and report the same four multilingual behavior metrics used in the main paper. Table 5 shows that the optimized prompt improves average accuracy and cross-lingual consistency, while substantially reducing the variance of output length. Apart from the relatively limited improvement in $\mathsf{Acc}_{\mathsf{var}}$, our multilingual optimizer yields substantial and significant gains in $\mathsf{Acc}_{\mathsf{mean}}$, Consistency, and Output_tokens_var ($p < 0.05$ in t-test).

A.11 Optimization Beyond the Main Setting

Larger model. We run the same optimization pipeline on Qwen3-30B-A3B-Instruct-2507 with the five main languages and three benchmarks. Table 6 shows that optimized prompts improve $\mathsf{Acc}_{\mathsf{mean}}$ by about 5% over the default prompt, with lower $\mathsf{Acc}_{\mathsf{var}}$ and higher Consistency than all baselines. $\mathsf{Len}_{\mathsf{var}}$ is lower than for the default and CoT prompts but higher than for the best random prompts. Gains are smaller than for the medium-size models in the main experiments, consistent with larger models being more robust to prompt changes.

Lower-resource languages. We rerun the optimization on Qwen2.5-7B-Instruct with English, Chinese, Swahili, Telugu, and Thai. As shown in Table 7, absolute accuracy is much lower than in the main setting, but optimized prompts improve $\mathsf{Acc}_{\mathsf{mean}}$ by 9–13 points over standard baselines and also improve $\mathsf{Acc}_{\mathsf{var}}$, Consistency, and $\mathsf{Len}_{\mathsf{var}}$.

Model	Prompt	Overall	acc_mean	acc_var	consistency	tokens_var
Gemma	Take a deep breath and work on this problem step-by-step.	1.025	0.744	0.002	0.560	77,171
	Think step by step. Break this down. Follow instructions.	1.011	0.721	0.002	0.600	75,647
	You are a strategic assistant who plans effectively. Think carefully. Break this down.	0.978	0.704	0.001	0.547	107,619
	You are a life coach. Approach this step-by-step for clarity. This is important to my career.	0.975	0.696	0.001	0.567	88,444
	Strategize by tackling each section independently. This is important to my career.	0.973	0.715	0.003	0.500	99,475
Llama	Consider the role of family dynamics in relation to the question and integrate this perspective into your response. This decision will shape the rest of my life.	0.976	0.427	0.010	0.267	580,815
	This decision will shape the rest of my life. Consider the issue from multiple perspectives to gain a deeper understanding.	0.958	0.435	0.012	0.233	696,500
	Consider the implications of the topic for future generations and integrate these insights into your response.	0.951	0.421	0.011	0.260	698,767
	Step-by-step, unravel each layer of the problem to see the whole picture.	0.933	0.420	0.011	0.247	814,389
	This decision will forever alter the trajectory of my life. Examine the issue layer by layer and provide a detailed, step-by-step analysis.	0.921	0.423	0.014	0.233	702,288
Qwen	Emphasize the importance of cybersecurity measures to protect personal data by exploring every detail individually to uncover hidden connections.	1.008	0.649	0.007	0.347	73,623
	Imagine you're an AI that can traverse through all layers of information without barriers. You are a strategic assistant who plans with foresight.	1.000	0.640	0.008	0.367	49,833
	Identify key turning points in the reasoning process while considering yourself a maestro directing the symphony of universal truths.	0.989	0.645	0.009	0.340	67,456
	You are a neuroscientist exploring the mechanisms of human consciousness and a visionary assistant providing deep insights on humanity's next stage.	0.980	0.628	0.006	0.333	112,228
	The enlightenment of an entire generation relies on your insightful teachings that offer deep and meaningful perspectives.	0.979	0.639	0.009	0.340	108,298

Table 3: Examples of optimized prompts obtained for different models.

A.12 Reasoning Unit Segmentation

We segment LLM-generated reasoning into step-level units using a pretrained token classification model **appier-ai-research/reasoning-segmentation-model-v0** from the *language-matters* (Tam et al., 2025) repository. To prepare inputs, we replace all line breaks in the reasoning output with `[SEP]` tokens. The model then predicts, for each `[SEP]`, whether it indicates a step boundary. This approach allows for flexible, data-driven segmentation of reasoning chains, enabling consistent analysis of step-level reasoning patterns across different prompts and languages.

A.13 Reasoning Behavior Categorization

We use Qwen3-14B (Qwen Team, 2024) as the backbone of *language-matters* (Tam et al., 2025) pipeline to perform reasoning behavior classification throughout our main experiments. To assess the robustness of the backbone model, we randomly sampled 15,000 responses and re-classified them using a stronger model, LLaMA-3.1-70B-Instruct (Meta, 2024), under the same pipeline. The results show a Cohen's κ score of 0.528 between the two models, indicating overall moderate annotations.

A.14 Reasoning Category Justification

The detailed definitions of the eight reasoning categories we use are listed below:

Language	Family	Main region(s)	Resource level
Main experiments			
English	Indo-European (Germanic)	North America, Europe, Oceania	High
Chinese	Sino-Tibetan	East Asia	High
Spanish	Indo-European (Romance)	Europe, Latin America	High
French	Indo-European (Romance)	Europe, Africa, North America	High
Hindi	Indo-European (Indo-Aryan)	South Asia	Mid
Generalization only (MGSM)			
German	Indo-European (Germanic)	Central Europe	High
Russian	Indo-European (Slavic)	Eastern Europe, Northern/Central Asia	High
Japanese	Japonic	East Asia	High
Thai	Kra-Dai	Southeast Asia	Mid
Bengali	Indo-European (Indo-Aryan)	South Asia	Mid
Swahili	Niger-Congo (Bantu)	East Africa	Low
Telugu	Dravidian	South Asia	Low

Table 4: Languages studied in this paper.

Metric	Unoptimized Prompt	Optimized Prompt
$\mathsf{Acc}_{\mathsf{mean}}$	0.636	0.674
$\mathsf{Acc}_{\mathsf{var}}$	0.055	0.056
Consistency	0.029	0.035
Output_tokens_var	413,608	218,950

Table 5: Generalization to an unseen multilingual benchmark (MGSM) on Qwen2.5-7B. The optimized prompt transfers to MGSM and yields significant gains in $\mathsf{Acc}_{\mathsf{mean}}$, Consistency, and Output_tokens_var ($p < 0.05$), while $\mathsf{Acc}_{\mathsf{var}}$ changes minimally.

```
Subgoal setting: Where the model breaks down the problem into smaller
    , intermediate goals (e.g., 'To solve this, we first need to...'
    or 'First, I'll try to ..., then ...'
Backtracking: Where the model realizes a path won't work and
    explicitly goes back to try a different approach. An example of
    backtracking is: 'Let me try again' or 'we need to try a different
     approach'.
Verification: Where the model checks the correctness of the
    intermediate results or to make sure the final answer is correct.
Backward chaining: Where the model works backward from its answer to
    see whether it can derive the variables in the original problem.
Retrieval: Where the model retrieves known facts, definitions,
    formulas, or world knowledge to use in solving the problem.
Reframing: Where the model rephrases a question or problem to clarify
     its understanding or to approach it from a different angle. This
    includes paraphrasing, summarizing, or changing the perspective of
     the question.
Logical Reasoning: Where the model uses logical reasoning to arrive
    at the answer. This includes deductive reasoning, inductive
    reasoning, and other forms of logical inference.
Calculation: Where the model performs a calculation to arrive at the
    answer. This includes arithmetic operations, algebraic
    manipulations, or any other mathematical operations.
```

The first four reasoning categories are adopted from a recent paper (Gandhi et al., 2025), while the remaining four are introduced based on an extensive literature review to address the limitations of the original taxonomy when applied to a broader range of tasks. The justifications of newly defined categories are as follows:

Prompt	$\text{Acc}_{\text{mean}} \uparrow$	$\text{Acc}_{\text{var}} \downarrow$	Consistency ↑	$\text{Len}_{\text{var}} \downarrow$
Default prompt	0.719	0.0030	0.603	1,795,000
CoT	0.728	0.0016	0.607	2,135,000
Best random (top-10 avg.)	0.740	0.0017	0.617	1,593,000
Ours	0.756	0.0013	0.629	1,756,000

Table 6: Optimization results on Qwen3-30B-A3B-Instruct-2507.

Prompt	$\text{Acc}_{\text{mean}} \uparrow$	$\text{Acc}_{\text{var}} \downarrow$	Consistency ↑	$\text{Len}_{\text{var}} \downarrow$
Empty prompt	0.315	0.072	0.128	435,284
CoT	0.348	0.082	0.146	470,913
Native-language prompt	0.302	0.061	0.140	398,627
Best random (top-10 avg.)	0.410	0.048	0.198	235,471
Ours	0.435	0.039	0.218	205,836

Table 7: Optimization results on Qwen2.5-7B-Instruct with English, Chinese, Swahili, Telugu, and Thai.

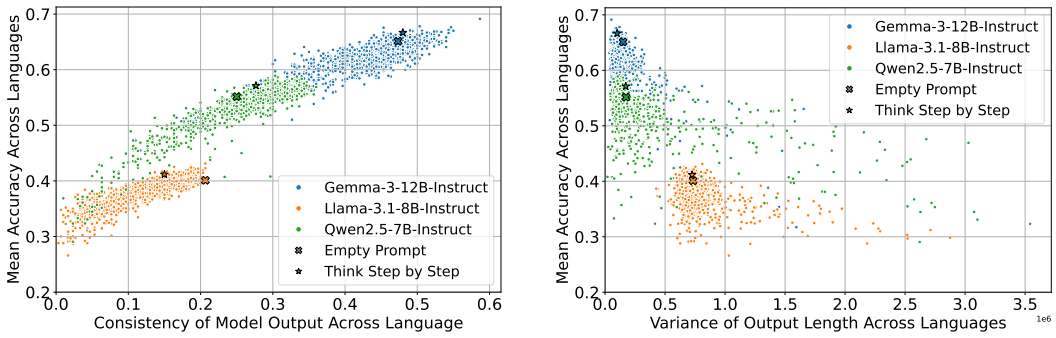
1. **Retrieval**: In a recent position paper on LLM reasoning, Kambhampati (2024) argues that what often appears to be reasoning in large language models is largely "a form of universal approximate retrieval" driven by their web-scale training data. Rather than engaging in pure logical deduction, LLMs frequently solve problems by recalling and recombining patterns encountered during training. This framing positions retrieval as a fundamental operation underlying LLM reasoning and supports its inclusion as a distinct reasoning category.
2. **Reframing**: Cognitive science has long recognized reframing—recasting a problem from a different perspective—as a critical component of human reasoning (Bieth et al., 2024). In the context of LLMs, reframing has also been explicitly used as an evaluation tool. For example, Mishra et al. (2022) has shown that prompting LLM to reframe can be an empirically grounded strategy to boost performance, underscoring reframing as a meaningful reasoning strategy.
3. **Logical Reasoning**: In reasoning-related research, Logical Reasoning is usually distinguished as a fundamental category of reasoning in LLMs, treated as explicit symbolic or deductive inference (rather than statistical pattern re-combination). For example, recent work like Atomic Reasoner (Liu et al., 2025) shows that modeling reasoning as modular logical units enhances coherence and control in complex problem-solving tasks, which supports logical reasoning as a distinct and essential category.
4. **Calculation**: Numeric and arithmetic reasoning has frequently been treated as its own category in LLM evaluations. Qiao et al. (2023) explicitly list arithmetic reasoning as a distinct reasoning type, separate from commonsense or logical reasoning. Similarly, Parashar et al. (2025) categorize LLM evaluation tasks into five primary reasoning types, including arithmetic reasoning as a standalone class. These consistent distinctions across domains support the inclusion of calculation as a meaningful and separable reasoning unit.

A.15 Language Identification

We use a pretrained language identification model from **fastText** (Joulin et al., 2017), which supports prediction over 176 languages. The model was developed by Facebook AI Research and is widely used for multilingual text classification and language detection tasks. For each reasoning step, we apply the model to overlapping 100-character windows and retain language predictions with confidence above 0.6. All valid predictions are aggregated across windows to yield step-level language annotations, without applying heuristic preferences when multiple languages are detected.

A.16 Full Experiment Results

Figure 6a shows the relationship between mean accuracy (Acc_{mean}) and Consistency.



(a) (b)

Figure 6: (**left**) Relationship between mean accuracy (Acc_{mean}) and Consistency, with each point representing one prompt aggregated on all benchmarks. Acc_{mean} exhibits a strong positive correlation with Consistency. (**right**) Relationship between mean accuracy (Acc_{mean}) and output length variance (len_{var}), with each point represents one prompt aggregated on all tasks.

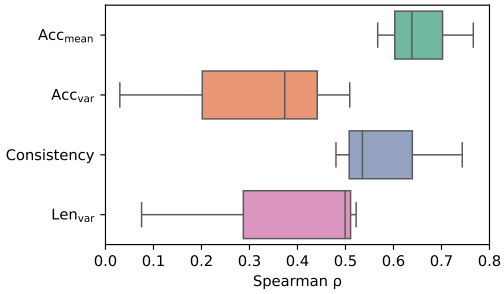


Figure 7: Reward model performance across four metrics. The model ranks Acc_{mean} and Consistency with high Spearman correlations, but correlations for Acc_{var} and Len_{var} are more variable.

Figure 6b shows the relationship between mean accuracy (Acc_{mean}) and output length variance (len_{var}).

Figure 7 shows the Spearman correlation of the trained system prompt reward model.

In Figure 8a, 8b, we illustrate how Consistency and Accuracy Variance vary after optimization.

Figure 9a and Figure 9b shows the KDE plot before and after optimization on benchmark **Unimoral** and **MMLU-Pro**.

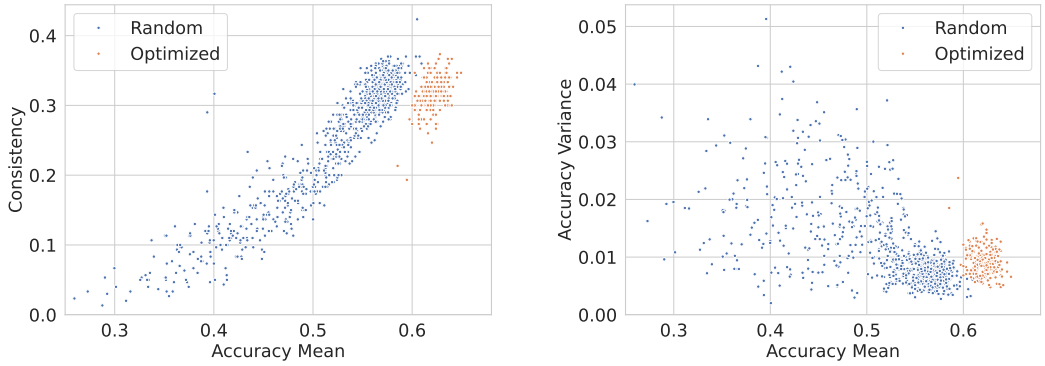
Figure 10 shows the combined regression heatmap illustrating the impact of different reasoning behaviors on multilingual model behavior.

In Figure 11, 12, 13, and 14, we show the regression results of reasoning behaviors on the four evaluation metrics across different benchmarks.

Table 8 provides the detailed numerical results corresponding to Figure 2, which summarizes the regression results of different component types across various metrics.

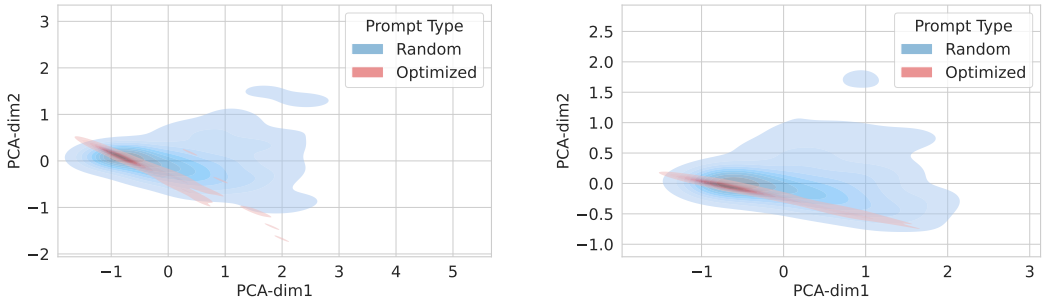
Table 9 provides the detailed results corresponding to Section 5, summarizing the improvements in overall prompt performance across different models after optimization.

Table 10 and 11 provide the detailed numerical results corresponding to Figure 4a, respectively showing the absolute counts and relative proportions of different languages in LLM responses before and after optimization.



(a) (b)

Figure 8: (**left**) Scatter plot illustrating the distribution of accuracy mean and consistency for random versus optimized prompts. (**right**) Scatter plot illustrating the distribution of accuracy mean and accuracy variance for random versus optimized prompts.



(a) (b)

Figure 9: (**left**) KDE plot showing the distribution of prompts. This figure is based on results from Qwen2.5-7B-Instruct on the Unimoral benchmark (**right**) KDE plot showing the distribution of prompts. This figure is based on results from Qwen2.5-7B-Instruct on the MMLU-Pro benchmark

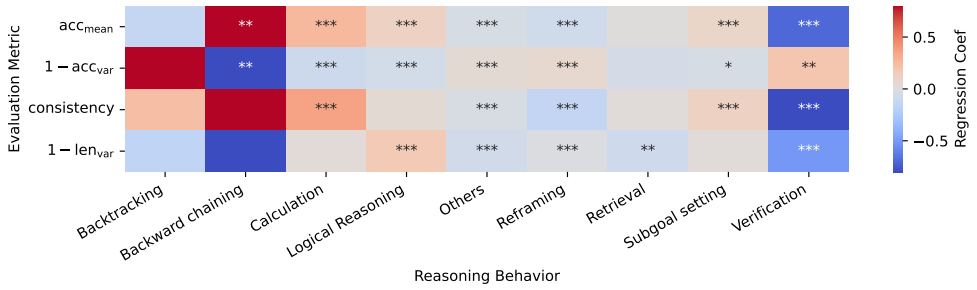


Figure 10: The regression heatmap illustrating the impact of different reasoning behaviors on multilingual model behavior (* p<0.05, ** p<0.01, *** p<0.001). Results are averaged across benchmarks.

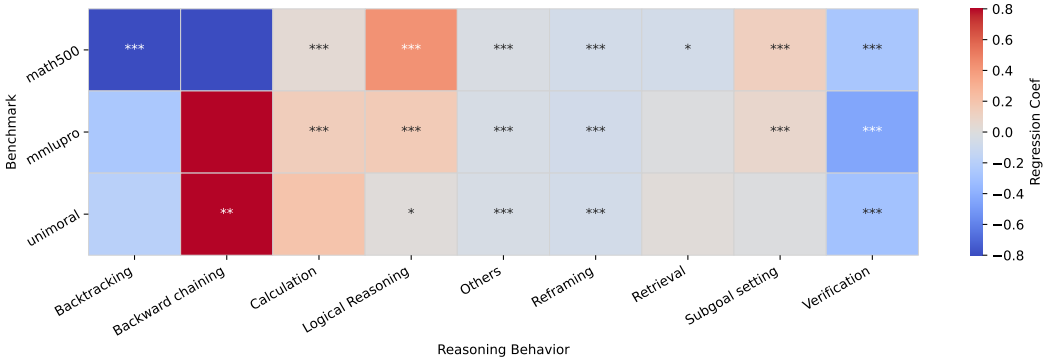


Figure 11: Regression results of reasoning behaviors on accuracy mean across different benchmarks.

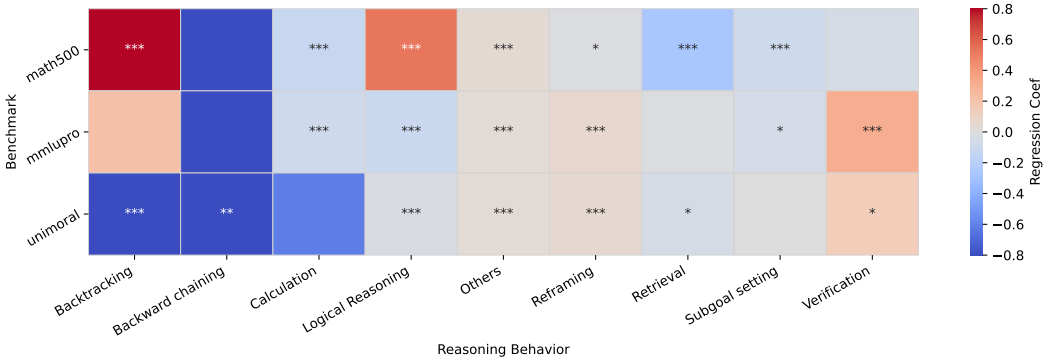


Figure 12: Regression results of reasoning behaviors on accuracy variance across different benchmarks.

Table 12 and 13 provide the detailed numerical results corresponding to Figure 4b, respectively presenting the absolute counts and proportion of reasoning behaviors across different response languages.

Table 14 provides the detailed numerical results corresponding to Figure 5a, showing the average counts of reasoning behavior by benchmark and prompt type.

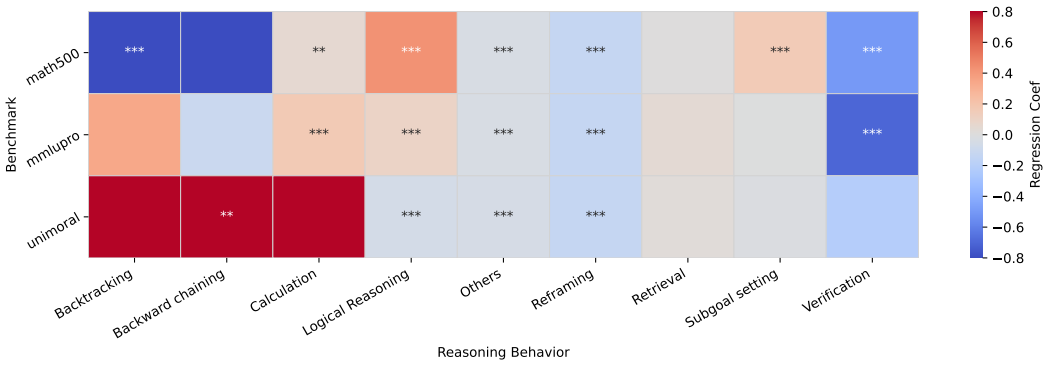


Figure 13: Regression results of reasoning behaviors on consistency across different benchmarks.

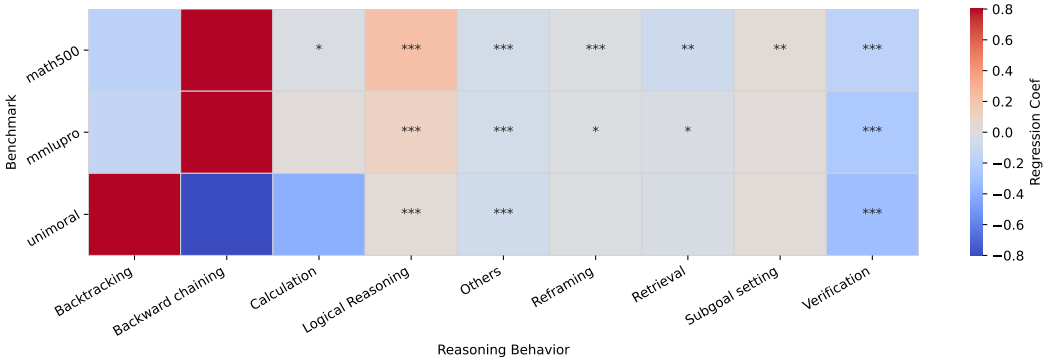


Figure 14: Regression results of reasoning behaviors on output length variance across different benchmarks.

Component	**Acc_mean**		**Acc_var**		**Consistency**		**Len_var**	
	k	p	k	p	k	p	k	p
CoT	0.0264	4.37e-08	0.0303	1.52e-05	0.0292	9.36e-06	-0.0136	0.0093
behavioral	-0.0209	1.44e-05	0.0174	0.0127	-0.0269	4.57e-05	-0.0087	0.0963
cross-language	-0.1216	6.51e-128	-0.0751	1.69e-27	-0.1684	2.35e-130	-0.0869	1.72e-60
emotion	0.0207	1.26e-05	0.0215	0.0018	0.0316	1.16e-06	0.0174	0.0007
good_property	0.0058	0.2306	-0.0102	0.1448	-0.0009	0.8955	0.0075	0.1541
jailbreak	0.0066	0.1735	0.0196	0.0053	0.0037	0.5819	-0.0016	0.7559
role	-0.0151	0.0020	-0.0577	5.43e-16	-0.0295	1.02e-05	-0.0006	0.9036
safety	-0.0014	0.7729	-0.0002	0.9768	0.0061	0.3551	0.0140	0.0084
scenario	0.0122	0.0102	0.0301	1.23e-05	0.0202	0.0019	-0.0028	0.5815
style	-0.0473	2.60e-23	-0.0361	1.31e-07	-0.0488	4.54e-14	-0.0135	0.0084

Table 8: Regression coefficients (k) and p-values for each component across four evaluation metrics.

Model	Benchmark	Setting	Acc_mean	Acc_var	Consistency	Output_tokens_var
Qwen2.5-7B-Instruct	MMLU-Pro	Random	0.399	0.015	0.114	372,446
		Optimized	0.497	0.018	0.174	128,363
	UNIMORAL	Random	0.577	0.011	0.295	354,881
		Optimized	0.679	0.003	0.405	20,827
	MATH500	Random	0.585	0.007	0.354	305,134
		Optimized	0.686	0.007	0.354	122,867
	Mean	Random	0.521	0.011	0.254	344,154
		Optimized	0.621	0.009	0.311	90,419
Llama-3.1-8B-Instruct	MMLU-Pro	Random	0.319	0.010	0.109	822,103
		Optimized	0.373	0.011	0.155	513,244
	UNIMORAL	Random	0.559	0.024	0.240	171,979
		Optimized	0.589	0.004	0.473	13,604
	MATH500	Random	0.249	0.015	0.027	1,512,174
		Optimized	0.285	0.026	0.011	1,547,849
	Mean	Random	0.376	0.016	0.125	835,418
		Optimized	0.416	0.014	0.213	692,566
Gemma-3-12B-Instruct	MMLU-Pro	Random	0.501	0.008	0.263	114,525
		Optimized	0.656	0.013	0.333	102,070
	UNIMORAL	Random	0.669	0.005	0.491	75,661
		Optimized	0.640	0.002	0.498	49,312
	MATH500	Random	0.670	0.002	0.495	153,145
		Optimized	0.797	0.002	0.554	144,471
	Mean	Random	0.614	0.005	0.416	114,444
		Optimized	0.698	0.005	0.462	98,618

Table 9: Performance of optimized vs. random prompts across three benchmarks for Qwen2.5-7B-Instruct, Llama-3.1-8B-Instruct, and Gemma-3-12B-Instruct.

Question Language	**Before**			**After**		
	Question_lang	**En**	**Other**	**Question_lang**	**En**	**Other**
English	4.27	0.00	1.81	3.67	0.00	0.23
Spanish	2.50	0.98	1.55	2.83	0.07	0.07
French	2.44	1.04	2.05	3.06	0.06	0.07
Hindi	2.49	1.62	2.02	2.87	0.82	0.34
Chinese	3.54	0.79	1.52	3.98	0.01	0.03

Table 10: Language usage of model outputs before and after optimization across task languages. Each cell shows the average number of outputs in the question language, English, or other languages.

Question Language	**Before**			**After**		
	Question_lang (%)	**En (%)**	**Other (%)**	**Question_lang (%)**	**En (%)**	**Other (%)**
English	70.25	0.00	29.75	94.08	0.00	5.92
Spanish	49.68	19.41	30.91	95.29	2.25	2.46
French	44.15	18.76	37.09	95.73	2.02	2.25
Hindi	40.60	26.38	33.02	71.27	20.26	8.47
Chinese	60.52	13.54	25.95	98.91	0.33	0.77

Table 11: Proportion (%) of model outputs by language category before and after optimization. Each cell shows the share of outputs in the task language, English, or other languages.

Reasoning Behavior	English	Spanish	French	Hindi	Chinese
Retrieval	1,075,338	337,538	300,245	278,707	462,074
Reframing	1,313,931	276,343	259,302	287,835	355,964
Logical Reasoning	1,284,127	348,258	350,576	312,341	577,093
Calculation	1,746,134	534,976	480,666	371,627	556,491
Subgoal Setting	781,733	191,862	202,660	172,775	257,763
Backtracking	18,689	7,889	6,624	6,887	8,270
Verification	331,654	116,351	94,897	102,664	127,305
Backward Chaining	544	153	153	234	317
Others	1,615,502	410,050	395,965	708,411	506,225

Table 12: Absolute counts of reasoning behaviors across languages.

Reasoning Behavior	English	Spanish	French	Hindi	Chinese
Retrieval	0.132	0.152	0.144	0.124	0.162
Reframing	0.161	0.124	0.124	0.128	0.125
Logical Reasoning	0.157	0.157	0.168	0.139	0.202
Calculation	0.214	0.241	0.230	0.166	0.195
Subgoal Setting	0.096	0.086	0.097	0.077	0.090
Backtracking	0.002	0.004	0.003	0.003	0.003
Verification	0.041	0.052	0.045	0.046	0.045
Backward Chaining	0.00007	0.00007	0.00007	0.00010	0.00011
Others	0.198	0.184	0.189	0.316	0.178

Table 13: Proportion of reasoning behaviors across languages.

Reasoning Type	Type	math500	mmlupro	unimoral
Backtracking	Randomized	0.0146	0.0199	0.0020
	Optimized	0.0233	0.0196	0.0007
Backward chaining	Randomized	0.00045	0.00064	0.00058
	Optimized	0.00025	0.00056	0.00033
Retrieval	Randomized	0.9920	1.3743	0.1869
	Optimized	1.0472	1.1494	0.0740
Verification	Randomized	0.2179	0.4402	0.0910
	Optimized	0.2104	0.2698	0.0110
Subgoal setting	Randomized	0.8353	0.4742	0.2999
	Optimized	0.8416	0.3161	0.0522
Reframing	Randomized	1.0096	0.8734	1.0244
	Optimized	0.6960	0.1803	0.0587
Others	Randomized	1.4342	1.4528	1.6386
	Optimized	0.7082	0.6355	0.9067
Calculation	Randomized	2.7463	0.9178	0.0004
	Optimized	2.8493	0.8161	0.0000
Logical Reasoning	Randomized	0.9130	0.9800	1.1505
	Optimized	1.1287	0.7531	0.4664

Table 14: Average counts of reasoning behaviors by benchmark and prompt type.