# Less Languages, Less Tokens: An Efficient Unified Logic Cross-lingual Chain-of-Thought Reasoning Framework

**Chenyuan Zhang**[1,6*] **Qiguang Chen**[2*] **Xie Chen**[5,6] **Zhuotao Tian**[1] **Bowen Xing**[4]
**Meishan Zhang**[1] **Libo Qin**[1,2,3†] **Baotian Hu**[1] **Min Zhang**[1]
[1] Harbin Institute of Technology, Shenzhen
[2] Central South University
[3] Text Computing and Cognitive Intelligence Ministry of Education Engineering Research Center, Guizhou University
[4] University of Science and Technology Beijing
[5] Shanghai Jiao Tong University
[6] Shanghai Innovation Institute
qinlibo@hit.edu.cn
cyzhang@stu.hit.edu.cn, charleschen2333@gmail.com

## Abstract

Cross-lingual chain-of-thought (XCoT) with self-consistency markedly enhances multilingual reasoning, yet existing methods remain costly due to extensive sampling of full trajectories across languages. Moreover, multilingual LLM representations vary strongly by language, hindering direct feature comparisons and effective pruning. Motivated by this, we introduce UL-XCoT, the first efficient unified logic cross-lingual reasoning framework that minimizes redundancy in token usage and latency, yielding the greatest efficiency under limited sampling budgets during inference. Specifically, UL-XCoT (1) achieves less languages by selecting, per query, a small candidate language set in a language-invariant unified logic space, (2) enables less tokens by monitoring logic-space trajectory dynamics during decoding to prune low-quality reasoning paths, and (3) aggregates the remaining high-quality trajectories via voting. Experiments on PolyMath across 18 languages and MMLU-ProX-Lite across 29 languages with DeepSeek-R1-Distill-Qwen-7B demonstrate that UL-XCoT achieves competitive accuracy while sharply cutting over 50% decoding token cost versus prior sampling baselines. UL-XCoT also delivers more stable gains on low-resource languages, underscoring consistently superior robustness where standard XCoT self-consistency method fails.

## 1 Introduction

Multilingual large language models (MLLMs) have shown strong reasoning and generalization abilities (Qin et al., 2025; Chen et al., 2024a; Lai et al., 2023; Resck et al., 2025), which Cross-lingual chain-of-thought (XCoT) can further optimize. Specifically, XCoT is a reasoning paradigm where inputs and intermediate steps use different languages (Qin et al., 2023; Zhang et al., 2024; Huang et al., 2026; Tran et al., 2025b), effectively activating their core reasoning capabilities (Ahuja et al., 2025; Chen et al., 2024b; Huang et al., 2023).

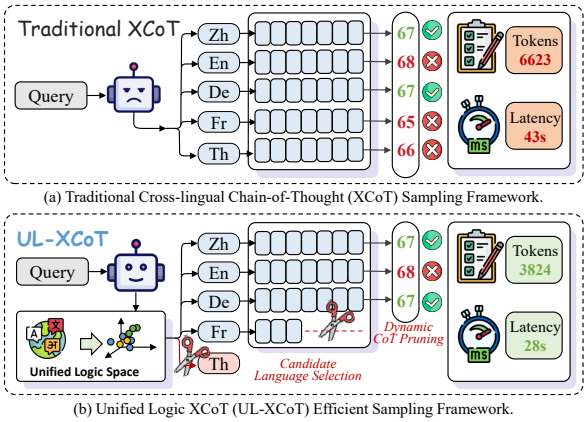



Figure 1: Traditional XCoT Sampling framework (a) generates complete reasoning trajectories with all languages (e.g., Chinese, English, German, French, and Thai). In contrast, Unified Logic XCoT (UL-XCoT) Efficient Sampling framework (b) uses the Unified Logic Mechanism for efficient language selection and selective trajectories generation.

As multilingual reasoning research advances, XCoT has attracted growing interest for test-time scaling via self-consistency, which samples multiple complete trajectories and aggregates outputs (Tran et al., 2025a; Khairi et al., 2025; Ghosh et al., 2025; Zhang et al., 2025b). Specifically, Qin et al. (2023) use voting across multiple XCoT trajectories to exploit cross-lingual complementarities. Zhang et al. (2024) optimize instructions to select matching languages and assign language-specific voting weights during aggregation. Ranaldi et al.

*Equal contribution.
†Corresponding author.



*Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29912–29929
July 2-7, 2026 

(2024) prompt the model to simulate multilingual experts, generate diverse trajectories, and derive answers via cross-referencing. Khairi et al. (2025) apply temperature scaling and Bayesian risk to sample and select high-quality XCoT trajectories.

While existing test-time scaling strategies on XCoT have improved reasoning consistency and performance, they remain limited by computational inefficiency. As shown in Figure 1(a), most prior approaches suffer from two drawbacks: (1) **full-language sampling**, which requires generating all candidate languages, and (2) **full-trace reasoning**, which requires generating all complete reasoning paths during inference. These issues cause redundant computation on ineffective languages or similar reasoning trajectories, leading inference costs to grow linearly with the number of languages and producing substantial redundant tokens.

Motivated by this, we propose the **Unified Logic Cross-lingual Chain-of-Thought (UL-XCoT) self-consistency framework**, which enhances XCoT efficiency via two modules: Candidate Language Selection (CLS) and Dynamic CoT Pruning (DCP). As shown in Figure 1 (b), UL-XCoT first utilizes a Unified Logic Mechanism (ULM) to establish a unified logical representation space to compare and filter reasoning processes across languages on a shared scale. It then applies CLS to evaluate candidate languages in this space and select a small subset **(less languages)** most relevant to the input query, reducing computation from irrelevant ones. Next, during reasoning, DCP tracks each language's CoT evolution and dynamically prunes redundant paths **(less tokens)** that are logically inconsistent. Finally, voting on the remaining high-quality cross-lingual reasoning paths cuts costs while preserving reasoning quality.

Experiments on PolyMath across 18 languages and on MMLU-ProX-Lite across 29 languages show that UL-XCoT performs an obvious accuracy-efficiency trade-off with DeepSeek-R1-Distill-Qwen-7B. On PolyMath, it attains competitive difficulty-weighted accuracy while consistently requiring the fewest generated tokens and the lowest latency across languages, reducing the average token count by more than 50% relative to AUTO-CAP and by more than 65% relative to SC. Beyond mathematical reasoning, UL-XCoT also generalizes well to MMLU-ProX-Lite, where it improves average accuracy while retaining clear efficiency advantages in both token usage and latency. Moreover, UL-XCoT yields stronger and more stable gains on a data-driven low-resource language subset, indicating greater robustness when standard prompting and sampling signals are weak.

Overall, the contributions of the paper are summarized as follows:

- We first point out an inherent efficiency limitation in the previous cross-lingual ensemble reasoning paradigm, where full-language enumeration implicitly assumes that the reasoning process in each language is equally important and must be fully generated, leading to substantial redundant computation.
- We introduce UL-XCoT, an efficient XCoT self-consistency framework improving inference efficiency from two dimensions: (1) less languages through efficient language selection, and (2) less tokens through dynamic pruning with early stopping during reasoning.
- Experimental results show that UL-XCoT significantly reduces inference cost while preserving accuracy, and yields particularly strong gains in low-resource languages.

For reproducibility, the code for this paper is available at https://github.com/chenyuanTKCY/UL-XCoT.

## 2 Method

In this section, we illustrate UL-XCoT. As illustrated in Figure 2, given an input query $x$ written in language $\ell$, UL-XCoT introduces a **Unified Logic Mechanism** that makes reasoning states across different languages comparable and measurable.

### 2.1 Overall Pipeline

Formally, let $\mathcal{L} = \{\ell_1, \ldots, \ell_M\}$ denote the set of all languages, $f_\theta$ an MLLM, and $\mathcal{A}$ the answer space. The overall inference process yields $\hat{a} = \text{UL-XCoT}_\theta(x|f_\theta) \in \mathcal{A}$. Specifically, this process comprises four stages:

- First, in order to achieve cross-lingual comparability of reasoning states in a unified logic space, we construct a unified logic mechanism via a projection operator $P^{(m)}_{\text{shared}}$ at layer $m$.
- Second, we select a candidate languages set $\mathcal{L}_{\text{par}}(x)$, guided by an understanding similarity score under the unified logic mechanism, thereby achieving less languages.
- Third, for each $\ell \in \mathcal{L}_{\text{par}}(x)$, we perform XCoT decoding with $f_\theta$, generating trajectory $x_\ell$ token by token in parallel. During decoding, a

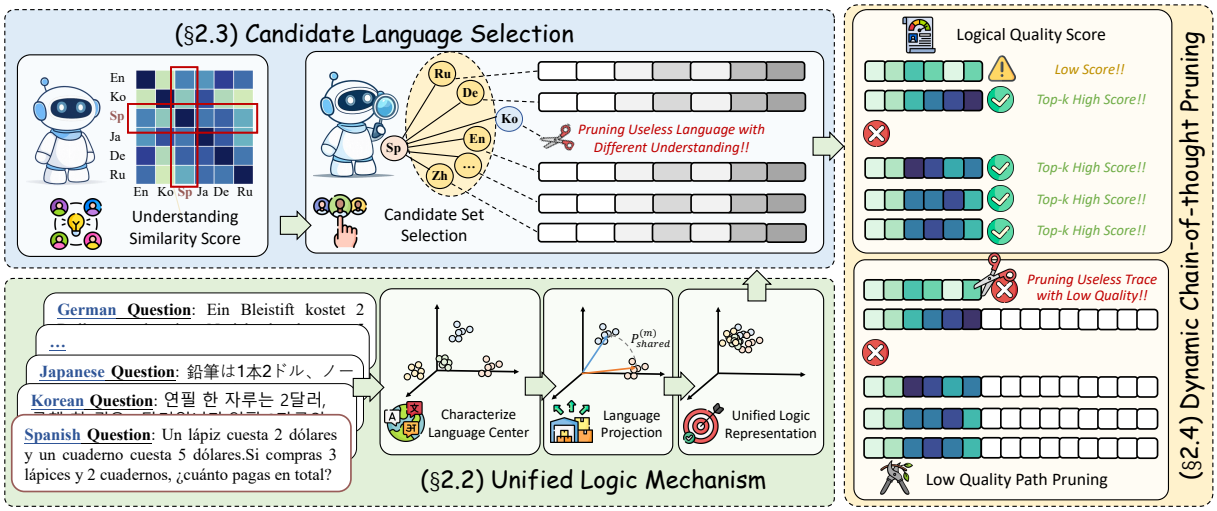



Figure 2: Overall framework of UL-XCoT, containing (i) The Unified Logic Mechanism, (ii) Candidate Language Selection, (iii) Dynamic Chain-of-Thought Pruning modules.

time-evolving confidence signal dynamically prunes low-quality paths, yielding a retained subset of languages $\mathcal{S}(x) \subseteq \mathcal{L}_{\text{par}}(x)$. The remaining trajectories are denoted by $S^*(x) = \{x_\ell : \ell \in \mathcal{S}(x)\}$. In this way, we can achieve less tokens during decoding.

- Finally, we extract an answer $a_\ell$ from each surviving trajectory and aggregate them via a voting operator $V$:

$$\hat{a} = V(\{a_\ell : \ell \in \mathcal{S}(x)\}). \tag{1}$$

Below, we describe: (1) the Unified Logic Mechanism (ULM) for constructing the unified logic space, (2) Candidate Language Selection (CLS) for forming $\mathcal{L}_{\text{par}}(x)$, and (3) Dynamic Chain-of-Thought Pruning (DCP) for obtaining $S^*(x)$.

## 2.2 The Unified Logic Mechanism

To fairly compare reasoning behaviors across languages, we need a unified representation space that suppresses language-specific surface variations while preserving task-relevant reasoning structures. We thus construct a **unified logic space**, where the signals act as decision criteria for CLS (Sec. 2.3) and DCP (Sec. 2.4).

Motivated by evidence that latent representations encode both linguistic structure and reasoning-relevant signals (Zhao et al., 2025; Hao et al., 2024; Feng et al., 2024), we use the transformation of embeddings for constructing the Unified Logic Mechanism. Let $H_m(x_\ell) \in \mathbb{R}^d$ denote the hidden state at Transformer layer $m$ for sample $x_\ell$ expressed under language $\ell \in \mathcal{L}$. To characterize systematic language-dependent shifts, we use a fixed validation set $\mathcal{X}_{\text{val}} = \{x^i_{\ell_i}\}_{i=1}^N$ and obtain language-specific realizations of the same content for each $\ell$ (e.g., by translation or consistent prompting). We then define a language center at layer $m$ as

$$\mu^{(m)}_\ell = \frac{1}{|\mathcal{X}_{\text{val}}|} \sum_{x^i_\ell \in \mathcal{X}_{\text{val}}} H_m(x^i_\ell). \tag{2}$$

Stacking centers across languages yields a multilingual shift matrix

$$M^{(m)} = \left[\mu^{(m)}_{\ell_1}, \mu^{(m)}_{\ell_2}, \ldots, \mu^{(m)}_{\ell_{|\mathcal{L}|}}\right] \in \mathbb{R}^{d \times |\mathcal{L}|}. \tag{3}$$

Following the intuition that cross-lingual variation concentrates on a low-dimensional subspace, we extract its principal directions via SVD:

$$M^{(m)} = U^{(m)} \Sigma^{(m)} V^{(m)\top}. \tag{4}$$

Let $r$ be a hyperparameter. We take the top-$r$ left singular vectors as an orthonormal basis of the language-variation subspace:

$$B^{(m)}_{\text{lang}} = U^{(m)}_{:,1:r}. \tag{5}$$

Its orthogonal complement defines the cross-lingually shared subspace, with projection operator

$$P^{(m)}_{\text{shared}} = I - \lambda B^{(m)}_{\text{lang}} B^{(m)\top}_{\text{lang}}. \tag{6}$$

For any input, we obtain its unified-logic-space representation by projection:

$$\tilde{H}_m(x_\ell) = P^{(m)}_{\text{shared}} h^{(m)}(x_\ell). \tag{7}$$

### 2.3 Candidate Language Selection

Candidate Language Selection (CLS) compares inputs across languages in the unified logic space for pre-screening. Since it requires no generation, hidden states in this space capture a unified understanding of the input. We compute an understanding similarity score for each language and select the top-$k$ as screening candidates.

**Understanding Similarity Score** To quantify cross-lingual understanding consistency, we define an understanding similarity score in the unified logic space. For input query $x_\ell$ in source language $\ell$ and each candidate target language $\ell' \in \mathcal{L}$, we construct a semantically equivalent rendition $x_{\ell'}$. At analysis layer $m = a$, we extract the last-token projected representations $\tilde{H}_a(x_\ell)$ and $\tilde{H}_a(x_{\ell'})$ in the unified logic space as the model's understanding states. We define the Understanding Similarity Score (USS) as:

$$\mathrm{USS}(x_\ell, x_{\ell'}) \triangleq \frac{\langle \tilde{H}_a(x_\ell),\, \tilde{H}_a(x_{\ell'}) \rangle}{\|\tilde{H}_a(x_\ell)\|_2 \, \|\tilde{H}_a(x_{\ell'})\|_2}. \quad (8)$$

This metric quantifies the preservation of identical understanding states across languages.

**Candidate Set Selection** We select the top-$k$ languages according to this score:

$$\mathcal{L}_{\text{par}}(x_\ell) = \text{Top-}k_{\ell' \in \mathcal{L}}\, \mathrm{USS}(x_\ell, x_{\ell'}). \quad (9)$$

It performs parallel XCoT sampling only over $\mathcal{L}_{\text{par}}(x_\ell)$. By enabling direct cross-lingual comparison in the unified logic space, this pre-filtering step is query-adaptive and reduces redundant tokens generated while preserving effective cross-lingual collaboration during inference.

### 2.4 Dynamic Chain-of-Thought Pruning

Dynamic Chain-of-Thought Pruning (DCP) monitors reasoning during XCoT decoding within the unified logic space and prunes low-quality paths online. This eliminates redundant generation from inconsistent or drifting trajectories, focusing computation on coherent high-quality paths.

**Logical Quality Score** We introduce a warm-up phase of length $T_{\text{warm}}$, during which pruning is disabled to avoid discarding paths prematurely. After warm-up, let $S_t$ denote the reasoning path from 0 to $t$-th step. For each path in $S_t$ with language $\ell$, we track its trajectory and compute a cohort score "$\texttt{score}^{(t)}(\cdot|\ell)$" from $t_0$-th step over a window of length $\tau$. The Logical Quality Score (LQS) is[1]:

$$\mathrm{LQS}(S_t|x_\ell, \ell') \triangleq \int_{t=t_0}^{t_0+\tau} \texttt{score}(S_t|x_\ell, \ell'). \quad (10)$$

**Low Quality Path Pruning.** At the end of the monitoring window ($T_E = T_{\text{warm}} + \tau + 1$), we collect the trace set $\mathcal{S}$ by retaining the top- $k'$ paths ranked by $\mathrm{LQS}(\ell)$:

$$\mathcal{S}(x_\ell) \;=\; \text{Top-}k'_{\ell' \in \mathcal{L}_{\text{par}}(x_\ell)}\, \mathrm{LQS}(S_t|x_\ell, \ell'), \quad (11)$$

where $k'$ follows pruning ratio $\rho$. Remaining paths terminate early, while retained paths continue decoding and aggregate via voting. This adaptively allocates compute to coherent paths, reducing redundancy and ensuring robustness.

## 3 Experiments

### 3.1 Experimental Setting

**Benchmark.** We mainly evaluate on PolyMath (Wang et al., 2025d), a multilingual mathematical reasoning benchmark with 18 parallel languages and 4 difficulty levels.

To test generalization, we further select a complementary benchmark on MMLU-ProX-Lite (Xuan et al., 2025), a multilingual multiple-choice benchmark spanning broader knowledge and reasoning categories.

**Evaluation protocol.** We assess both effectiveness and efficiency: (1) **Accuracy:** All methods use the same backbone and decoding constraints; for SC and AUTOCAP, we control the sample budget at UL-XCoT's worst-case level. We follow PolyMath in reporting DW-ACC. (2) **Efficiency:** We measure inference cost via generated tokens and wall-clock latency under identical conditions.

**Experimental setup.** We conduct all experiments with DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025). All runs are executed on NVIDIA RTX A6000 GPUs (48 GB). We set the maximum generation length in the interval of 2048–10240 according to PolyMath difficulty and stability of its performance. For prompting, we use a **concise-reasoning template**[2] for our method and baseline.

[1] Details are in Appendix A.

[2] The prompt is provided in Appendix D.

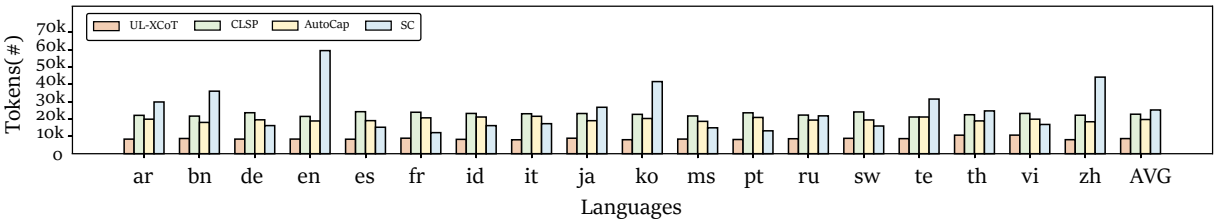


Figure 3: Average decoding token cost during generation on PolyMath.

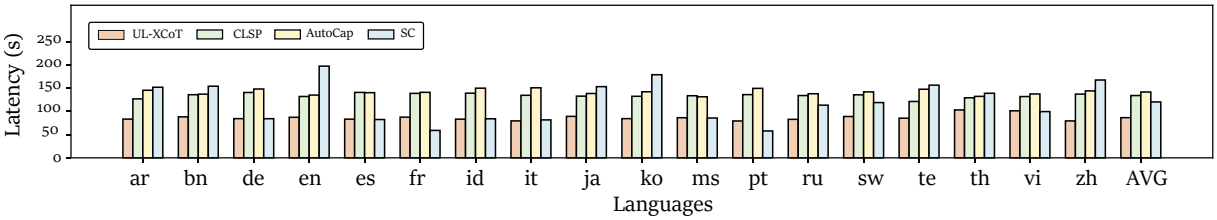


Figure 4: Average end-to-end latency across languages during generation on PolyMath.

**UL-XCoT configuration.** We set $\lambda = 0.4$, use layer $m = 13$ for logic-space representations (Section 2.3), set $|\mathcal{L}_{\text{par}}(x)| = 9$–$12$, and use a warm-up stage with $T_{\text{warm}} = 10$. After warm-up, we compute trajectory signals in a sliding window $\tau = 3c$ and adjust the pruning ratio $\rho$ within 20%–60% throughout decoding.

**Baselines.** We compare UL-XCoT with representative reasoning strategies spanning single-path prompting and sampling-based test-time scaling: (i) CLP (CLSP[3]) (Qin et al., 2023), a cross-lingual prompting framework that leverages multilingual signals to improve reasoning robustness; (ii) CoT (Wei et al., 2022), standard single-trajectory chain-of-thought prompting; (iii) SC (Wang et al., 2022), self-consistency that samples diverse CoT trajectories and aggregates the answers via majority voting; (iv) AUTOCAP (Zhang et al., 2024), an adaptive variant of SC that selects languages and assigns weights for aggregation; (v) ST-BoN (Wang et al., 2025c), an efficient sampling-based method that improves test-time scaling under a fixed budget in single-language inference; and (vi) UL-CoT, a monolingual counterpart of UL-XCoT featuring the identical DCP module.

## 3.2 Results for UL-XCoT

### 3.2.1 The Superior Efficiency.

Figures 3 & 4 compare the average decoding cost and end-to-end latency across languages.

**UL-XCoT can achieve marked efficiency.** As illustrated in Figure 3, UL-XCoT consistently uses the fewest tokens across all languages, achieving the lowest average token count with over 50% reduction relative to AUTOCAP and more than 65% relative to SC. This efficiency is mirrored in latency metrics (Figure 4), where UL-XCoT also exhibits the lowest latency in most languages[4].

**UL-XCoT can also reduce overthinking in high-resource languages.** As shown in Figure 3, some baselines (notably SC) exhibit pronounced spikes in token consumption on high resource languages like English and Chinese, suggesting occasional overthinking with excessively long reasoning traces. UL-XCoT largely avoids such extreme outliers, providing more steady inference-time behavior and significant measures toward cost for multilingual deployment.

### 3.2.2 The Comparable Performance.

Table 1 & 4 provide a comparative evaluation of UL-XCoT against a range of multilingual reasoning baselines on PolyMath[5].

**UL-XCoT achieves competitive accuracy.** Under the same evaluation setting, UL-XCoT delivers consistently strong performance across languages and four difficulty levels, ranking at or near the top on average. Table 4 shows competitive DW-ACC on PolyMath-Full across all difficulties, confirming effectiveness on challenging problems. Overall, UL-XCoT provides robust gains over prior prompting and sampling-based baselines for cross-lingual mathematical reasoning.

**UL-XCoT also shows robustness in low-resource languages.** We evaluate robustness on a data-driven low-resource subset (Bandarkar et al., 2024), performed as languages where both CoT and SC

[3]The Self-consistency version of CLP.

[4]A per-difficulty breakdown of token cost and wall-clock latency is provided in Appendix B.

[5]The overall performance is provided in Appendix C

Table 1: Performance on PolyMath across 18 languages and four difficulty levels from top to bottom.

| ACC | ar | bn | de | en | es | fr | id | it | ja | ko | ms | pt | ru | sw | te | th | vi | zh | AVG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PolyMath-Low | | | | | | | | | | | | | | | | | | | |
| CoT (Wei et al., 2022) | 52.0 | 42.4 | 56.0 | 83.2 | 72.8 | 56.8 | 59.2 | 68.0 | 41.6 | 42.4 | 55.2 | 67.2 | 71.2 | 4.0 | 13.6 | 49.6 | 59.2 | 76.8 | 54.0 |
| CLP (Qin et al., 2023) | 39.2 | 32.8 | 45.6 | 58.4 | 48.0 | 47.2 | 38.4 | 46.4 | 40.8 | 32.8 | 47.2 | 52.0 | 36.0 | 40.8 | 32.8 | 45.6 | 40.8 | 40.8 | 42.5 |
| SC (Wang et al., 2022) | 68.8 | 59.2 | 68.8 | **88.8** | 77.6 | 76.8 | 73.6 | 78.4 | 64.8 | 64.0 | 72.0 | 73.6 | 79.2 | 12.0 | 26.4 | 66.4 | 70.4 | 81.6 | 66.8 |
| CLSP (Qin et al., 2023) | 77.6 | **83.2** | 81.6 | 81.6 | 80.8 | **84.8** | 82.4 | 81.6 | 82.4 | 80.8 | 78.4 | 81.6 | 80.0 | 80.0 | 80.8 | 83.2 | **84.0** | 80.0 | 81.4 |
| AUTOCAP (Zhang et al., 2024) | 80.0 | 81.6 | 81.6 | 76.8 | **84.0** | 81.6 | 80.8 | 79.2 | 81.6 | 74.4 | 84.8 | **83.2** | 81.6 | **84.0** | 80.8 | 78.4 | 83.2 | 79.2 | 80.9 |
| ST-BoN (Wang et al., 2025c) | 60.0 | 62.4 | 65.6 | 69.6 | 64.8 | 72.0 | 64.0 | 63.2 | 61.6 | 61.6 | 63.2 | 62.4 | 64.8 | 60.0 | 63.2 | 59.2 | 61.6 | 63.2 | 63.5 |
| UL-CoT | 68.0 | 55.2 | 68.0 | 86.4 | 79.2 | 76.0 | 71.2 | 79.2 | 63.2 | 62.4 | 73.6 | 75.2 | 73.6 | 9.6 | 24.0 | 61.6 | 72.0 | 80.0 | 65.5 |
| **UL-XCoT** | **81.6** | **83.2** | **84.0** | 84.8 | 82.4 | 84.0 | **85.6** | **83.2** | **84.8** | **84.0** | **85.6** | **83.2** | **85.6** | 83.2 | **84.0** | **84.0** | 83.2 | **82.4** | **83.8** |
| PolyMath-Medium | | | | | | | | | | | | | | | | | | | |
| CoT (Wei et al., 2022) | 18.4 | 12.8 | 18.4 | 21.6 | 20.0 | 14.4 | 15.2 | 16.8 | 14.4 | 20.0 | 13.6 | 16.8 | 18.4 | 8.8 | 8.0 | 14.4 | 19.2 | 24.8 | 16.4 |
| CLP (Qin et al., 2023) | 17.6 | 17.6 | 16.8 | 18.4 | 18.4 | 20.0 | 15.2 | 17.6 | 20.8 | 19.2 | 16.0 | 17.6 | 18.4 | 14.4 | 14.4 | 8.8 | 21.6 | 16.8 | 17.2 |
| SC (Wang et al., 2022) | 30.4 | 22.4 | 27.2 | 38.4 | 25.6 | 24.8 | 22.4 | 24.0 | 19.2 | 32.0 | 17.6 | 24.8 | 32.0 | 12.0 | 12.8 | 24.0 | 29.6 | 39.2 | 25.5 |
| CLSP (Qin et al., 2023) | 32.0 | 28.8 | 33.6 | 29.6 | 28.0 | **32.8** | 31.2 | 25.6 | **31.2** | 28.8 | 30.4 | **28.8** | 28.8 | 32.0 | 30.4 | **29.6** | **33.6** | 32.0 | 30.4 |
| AUTOCAP (Zhang et al., 2024) | **35.2** | **34.4** | 30.4 | 24.0 | 31.2 | 29.6 | 25.6 | 26.4 | **31.2** | **36.0** | 29.6 | **28.8** | 27.2 | 30.4 | 28.0 | 25.6 | 28.8 | 26.4 | 29.4 |
| ST-BoN (Wang et al., 2025c) | 20.8 | 15.2 | 15.2 | 19.2 | 22.4 | 16.0 | 17.6 | 17.6 | 17.6 | 16.8 | 19.2 | 20.0 | 14.4 | 16.8 | 17.6 | 24.0 | 19.2 | 21.6 | 18.4 |
| UL-CoT | 28.0 | 25.6 | 26.4 | **45.6** | 26.4 | 24.0 | 22.4 | 25.6 | 21.6 | **36.0** | 17.6 | 19.2 | 32.0 | 13.6 | 15.2 | 27.2 | 24.0 | **43.2** | 26.3 |
| **UL-XCoT** | 28.8 | 28.0 | **35.2** | 32.0 | **35.2** | 29.6 | **33.6** | **28.8** | 27.2 | 27.2 | 31.2 | 28.0 | **32.8** | **38.4** | **33.6** | **29.6** | 32.8 | 28.0 | **31.1** |
| PolyMath-High | | | | | | | | | | | | | | | | | | | |
| CoT (Wei et al., 2022) | 5.6 | 6.4 | 10.4 | 8.0 | 10.4 | 8.0 | 4.0 | 9.6 | 7.2 | 8.0 | 4.8 | 8.8 | 8.0 | 0.8 | 1.6 | 7.2 | 8.8 | 12.8 | 7.2 |
| CLP (Qin et al., 2023) | 7.2 | 8.0 | 8.8 | 7.2 | 6.4 | 10.4 | 8.8 | 7.2 | 12.0 | 4.8 | 9.6 | 10.4 | 8.0 | 8.0 | 6.4 | 6.4 | 6.4 | 6.4 | 7.9 |
| SC (Wang et al., 2022) | 14.4 | 11.2 | 9.6 | **20.0** | 12.0 | 12.0 | 8.0 | 13.6 | 8.8 | **15.2** | 8.0 | 9.6 | 10.4 | 2.4 | 4.0 | 11.2 | 9.6 | 20.8 | 11.2 |
| CLSP (Qin et al., 2023) | **15.2** | 13.6 | **15.2** | 15.2 | **16.0** | 15.2 | 16.0 | **15.2** | **16.8** | 13.6 | **15.2** | **16.8** | **16.0** | **16.8** | 12.8 | 11.2 | 14.4 | 11.2 | **14.8** |
| AUTOCAP (Zhang et al., 2024) | 13.6 | 13.6 | 13.6 | 14.4 | 12.8 | 20.0 | 16.8 | 14.4 | **16.8** | 12.8 | 12.0 | 14.4 | 14.4 | 12.0 | **13.6** | 12.8 | **15.2** | 12.8 | 14.2 |
| ST-BoN (Wang et al., 2025c) | 12.0 | 10.4 | 8.8 | 3.2 | 7.2 | 6.4 | 8.0 | 14.4 | 7.2 | 8.8 | 10.4 | 8.0 | 9.6 | 10.4 | 7.2 | 8.8 | 8.0 | 8.0 | 8.7 |
| UL-CoT | 12.0 | **16.0** | 10.4 | **29.6** | 14.4 | 7.2 | 7.2 | 12.0 | 8.8 | 17.6 | 8.0 | 10.4 | 13.6 | 4.8 | 4.0 | 11.2 | 8.8 | **25.6** | 12.3 |
| **UL-XCoT** | 13.6 | 11.2 | 12.8 | 11.2 | 14.4 | 15.2 | **17.6** | 13.6 | 12.8 | 11.2 | 12.8 | 14.4 | 15.2 | 14.4 | 12.0 | **16.8** | 12.8 | 12.0 | 13.6 |
| PolyMath-Top | | | | | | | | | | | | | | | | | | | |
| CoT (Wei et al., 2022) | 5.6 | 1.6 | 7.2 | 2.4 | 7.2 | 9.6 | 7.2 | 3.2 | 8.8 | 3.2 | 4.8 | 8.0 | 5.6 | 0.8 | 0.8 | 6.4 | 4.8 | 4.0 | 5.1 |
| CLP (Qin et al., 2023) | 4.0 | 2.4 | 8.0 | 9.6 | 5.6 | 8.8 | 5.6 | 8.0 | 9.6 | 8.8 | 8.8 | 4.8 | 9.6 | 8.0 | 6.4 | 4.0 | 7.2 | 8.8 | 7.1 |
| SC (Wang et al., 2022) | **12.0** | 4.8 | 7.2 | 7.2 | 8.8 | 9.6 | 8.8 | **10.4** | 4.0 | 12.8 | 9.6 | 8.8 | 9.6 | 3.2 | 3.2 | 7.2 | 8.0 | 12.0 | 8.2 |
| CLSP (Qin et al., 2023) | 8.8 | 7.2 | **12.8** | 11.2 | 6.4 | 8.8 | **12.8** | 7.2 | **10.4** | **14.4** | 9.6 | 5.6 | 8.0 | 9.6 | **11.2** | **11.2** | 8.8 | 9.6 | 9.6 |
| AUTOCAP (Zhang et al., 2024) | 8.8 | 8.8 | 8.8 | 9.6 | 9.6 | 8.8 | 10.4 | 6.4 | 7.2 | 12.8 | **14.4** | **9.6** | 6.4 | 8.0 | 9.6 | 5.6 | 8.0 | 11.2 | 9.1 |
| ST-BoN (Wang et al., 2025c) | 7.2 | 6.4 | 5.6 | 8.8 | 5.6 | **10.4** | 8.8 | 7.2 | 10.4 | 8.8 | 9.6 | 9.6 | 7.2 | 5.6 | 7.2 | 10.4 | 10.4 | 7.2 | 8.1 |
| UL-CoT | 10.4 | 9.6 | 5.6 | **22.4** | **10.4** | **10.4** | 7.2 | 9.6 | 7.2 | **15.2** | 5.6 | 8.8 | 9.6 | 3.2 | 5.6 | 6.4 | 8.0 | **21.1** | 9.4 |
| **UL-XCoT** | 10.4 | **12.8** | 10.4 | 12.8 | 9.6 | **10.4** | 9.6 | 9.6 | 9.6 | 9.6 | 11.2 | 8.0 | **11.2** | **13.6** | 8.8 | **11.2** | **12.0** | 10.4 | **10.6** |

Table 2: Ablation study for 3 proposed modules on PolyMath-Low subset.

| Subset | ACC. | Latency (s) | Tokens (#) |
|---|---|---|---|
| UL-XCoT w/o CLS | 84.4 | 36.2 | 5560 |
| UL-XCoT w/o DCP | 81.4 | 30.7 | 3893 |
| UL-XCoT w/o ULM | 79.8 | 25.4 | 3098 |
| UL-XCoT w/o all modules | **85.2** | 35.9 | 7518 |
| UL-XCoT | 83.8 | **24.6** | **3092** |

fall below their mean DW-ACC across all languages. On this challenging set, UL-XCoT remains consistently strong performance, indicating more stable and reliable cross-lingual reasoning when language resources are limited.

**The multilingual gain does not collapse to pruning alone.** To separate the effect of multilingual collaboration from the gain brought by pruning, we further introduce a monolingual control variant, **UL-CoT**. This variant keeps the same unified logic representation and dynamic pruning signals as UL-XCoT, but samples and aggregates reasoning traces only within the query language. As shown in Table 1, UL-XCoT consistently outperforms UL-CoT across all four PolyMath difficulty levels,

indicating that the benefit of UL-XCoT does not come only from pruning more efficiently; multilingual interaction itself contributes substantial performance gains, especially on lower-resource languages.

### 3.2.3 The Robust Generalization.

**UL-XCoT transfers beyond PolyMath.** To complement the main PolyMath evaluation, we compare UL-XCoT and CLSP on MMLU-ProX-Lite (Xuan et al., 2025) under the same evaluation protocol. This benchmark covers a broader set of multilingual knowledge-and-reasoning multiple-choice questions, providing a task setting distinct from multilingual math reasoning. As shown in Table 3, UL-XCoT improves the average accuracy from 40.5 to 43.6, outperforms CLSP in 19 of 29 languages, and ties in 2 languages. And the gains also extend to several lower-resource languages, suggesting that the method can generalize across multilingual reasoning tasks.

**UL-XCoT retains its efficiency advantage on**

Table 3: Additional benchmark results on MMLU-ProX-Lite across 29 languages.

| Languages | af | ar | bn | cs | de | en | es | fr | hi | hu | id | it | ja | ko | mr | ne | pt | ru | sr | sw | te | th | uk | ur | vi | wo | yo | zh | zu | AVG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CLSP (Qin et al., 2023) | 37.3 | **44.1** | 32.2 | **44.1** | **42.4** | 39.0 | 40.7 | **42.4** | 35.6 | 37.3 | **47.5** | 37.3 | 39.0 | **44.1** | **42.4** | 35.6 | **42.4** | 44.1 | 40.7 | 39.0 | 37.3 | 42.4 | 37.3 | 37.3 | **45.8** | 42.4 | 44.1 | **42.4** | 39.0 | 40.5 |
| **UL-XCoT** | **40.7** | 39.0 | **52.5** | 40.7 | 40.7 | **49.2** | **52.5** | 40.7 | **39.0** | **42.4** | 44.1 | **45.8** | **45.8** | 39.0 | 40.7 | **42.4** | **42.4** | **45.8** | **45.8** | **42.4** | **40.7** | **45.8** | **49.2** | **39.0** | 42.4 | **45.8** | **45.8** | **42.4** | **42.4** | **43.6** |

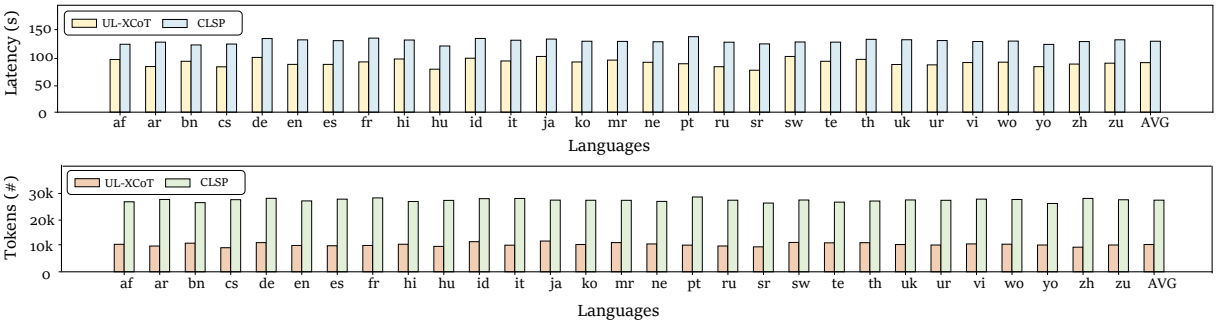



Figure 5: Average token cost and end-to-end latency across languages during generation on MMLU-ProX-Lite.

**the new benchmark.** Figure 5 reports the generated tokens and end-to-end latency across the 29 languages. UL-XCoT reduces average token usage from 27,679.3 to 10,543.6 and average latency from 134.2 s to 93.7 s. This result is consistent with our main findings: candidate language selection and dynamic pruning avoid unnecessary multilingual sampling, yielding a better cost–quality trade-off than CLSP even on a different task.

## 3.3 Analysis

### 3.3.1 Effectiveness of Each Module

To quantify the contribution of each module, we carry out ablation studies on PolyMath-Low by removing one module at a time [6] in Table 2.

**ULM is the main accuracy contributor under a matched compute budget.** A key observation from Table 2 is that UL-XCoT and UL-XCoT w/o ULM operate under nearly identical compute budgets (24.6s/3092 tokens vs. 25.4s/3098 tokens). Therefore, the accuracy drop after removing ULM reflects a genuine algorithmic gain rather than an artifact of increased sampling or longer decoding. Mechanistically, ULM maps language-specific chains into a unified, comparable logic space, enabling reliable cross-lingual confidence accumulation and coherent trajectory selection without increasing tokens and latency.

**CLS primarily improves efficiency by reducing the search space.** CLS mainly improves efficiency by proactively shrinking the candidate language set to a few logically relevant participants,

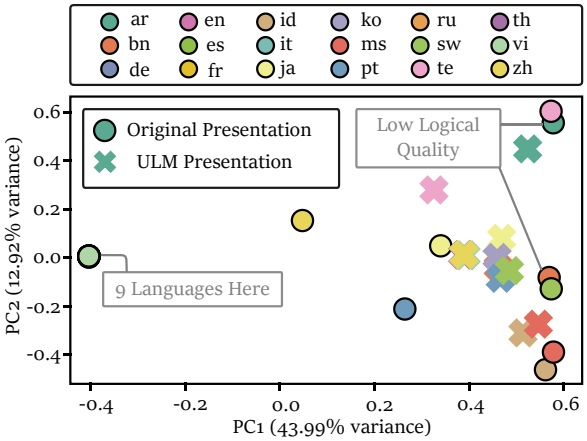



Figure 6: PCA projection of same-query embedding representations across 18 languages. Circles denote the original representations, while crosses indicate ULM-transformed representations in the unified logic space.

reducing exploration over noisy or irrelevant paths. As shown in Table 2, removing CLS enlarges the active set and leads to higher latency and tokens, indicating that CLS saves compute by avoiding low-value multilingual trajectories early.

**DCP saves compute via online pruning of low-quality trajectories.** DCP further reduces computation through early truncation: it monitors trajectory quality online and stops paths that show clear signs of low utility. And they do not continue consuming the full decoding budget. This strategy shrinks the active set early and focuses computation on the remaining competitive candidates during inference time. From Table 2, removing DCP substantially increases token usage and latency, while yielding only limited gains in final performance. These results suggest that many additional steps generated without DCP are largely redundant and rarely translate into better final votes.

[6] Variants without CLS or DCP may implicitly use more compute, which can inflate accuracy; hence accuracy should be read together with compute.

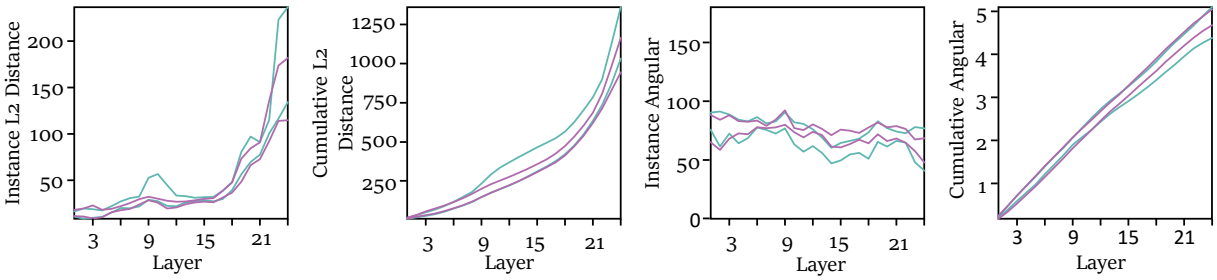



Figure 7: Layer-wise evolution of decoding embeddings measured by L2 distance and Angular across transformer layers. For each layer, we report the sample-wise extrema (min/max) to visualize the variation range. Purple denotes the model with ULM, while blue denotes the model without ULM.

#### 3.3.2 ULM can effectively unify cross-lingual logic representations.

To assess how ULM unifies cross-lingual logical representations, we analyze (1) static alignment across languages and (2) dynamic decoding trajectories under ULM.

**ULM disentangles static language-specific variation.** To visualize static alignment, we extract logic-space embeddings for the same query across 18 languages at a fixed decoding step and project them into 2D using PCA. Figure 6 shows that ULM of UL-XCoT disentangles surface-form variation, producing comparable, language-invariant representations across languages. After removing language-specific components, embeddings exhibit greater invariance: cross-lingual samples cluster tightly with consistent nearest neighbors. This indicates that the retained subspace encodes a shared logic state, not superficial differences.

**ULM can align dynamic reasoning trajectories.** Beyond static alignment, Figure 7 tracks how hidden-state geometry evolves across layers when answering the same queries in different languages.

With ULM, trajectories show a consistent distribution across languages, reflecting shared evolution patterns in the logic space. Without ULM, trajectories diverge markedly, with greater sensitivity and larger geometric gaps due to linguistic difference. Thus, ULM removes superficial variation, enabling DCP's chain-of-thought pruning via reliable geometric signals in the unified logic space.

#### 3.3.3 CLS can adaptively select appropriate languages without bias.

As shown in Figure 8, we count how often each language appears in the CLS-selected set $\mathcal{L}_{\text{par}}(x)$ across the full evaluation suite. CLS does not collapse to a single language: each language contributes roughly $3.7\% \sim 7.7\%$ of all selections

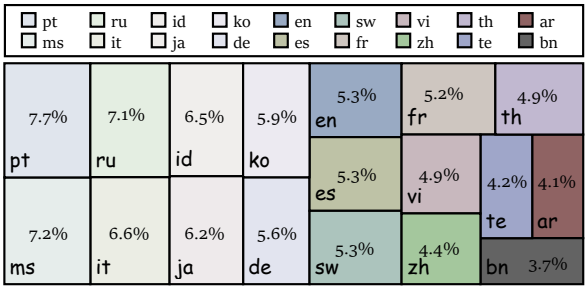



Figure 8: Distribution of languages selected by CLS, measured by the frequency of each language appearing in $\mathcal{L}_{\text{par}}(x)$ over the full evaluation suite.

(mean $\approx 5.6\%$), indicating broad coverage without a dominant bias. While a few languages (e.g., pt/ms/ru/it/id) are selected slightly more often, CLS still consistently includes lower-frequency languages (e.g., bn/ar/te/zh) for a non-trivial portion of inputs, reflecting query-adaptive selection rather than a fixed or heuristic language list.

#### 3.3.4 DCP enables quality-aware pruning for efficiency.

**DCP can effectively balance performance and efficiency.** To assess the impact of the pruning ratio $\rho$ in DCP, we vary $\rho$ from 0.0 to 0.9 and measure both accuracy and efficiency. As shown in Figure 9, for $\rho < 0.85$, higher $\rho$ slightly degrades accuracy but yields an almost linear reduction in latency and token usage, indicating that DCP removes low-confidence paths with minimal quality loss. Overall, a moderate $\rho$ of 0.55–0.70 provides the best trade-off, achieving substantial efficiency gains while largely preserving performance.

**DCP can truly prune low-quality paths.** To verify DCP's pruning quality, we analyze a high-performing subset where both Pruned and Full XCoT achieve strong overall performance. As

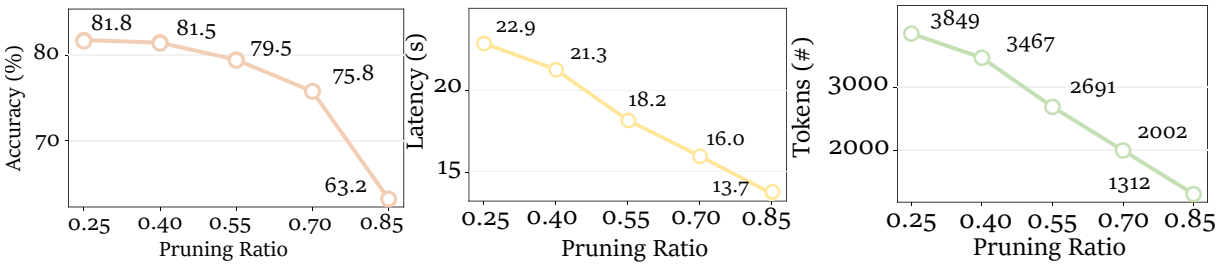



Figure 9: Impact of the pruning ratio $\rho$ on accuracy (left), latency (middle), and generated tokens (right).

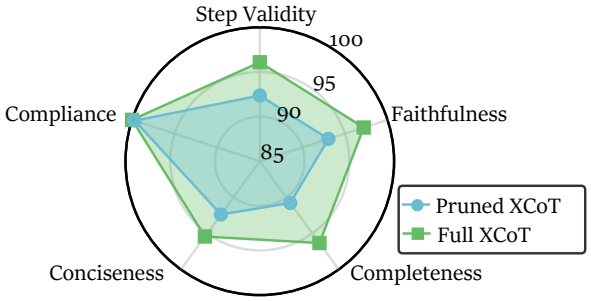



Figure 10: Quality comparison between trajectories pruned by DCP (Pruned XCoT) and those retained in Full XCoT, scored by an LLM judge over five criteria.

shown in Figure 10, we use an LLM-as-a-judge[7] to score trajectories (0–100) on step validity, faithfulness, completeness, conciseness, and compliance. Pruned trajectories consistently score lower, with the largest drops in step validity and completeness, indicating weaker logical coherence and more missing intermediate reasoning.

## 4 Related Work

### 4.1 Cross-lingual Chain-of-Thought Reasoning

Chain-of-Thought (CoT) prompting elicits explicit intermediate steps, improving reliability on multi-step problems (Wei et al., 2022; Kojima et al., 2022; Qin et al., 2026; Chen et al., 2025c). In cross-lingual settings, CoT can transfer across languages and strengthen in underrepresented languages as model scale grows (Shi et al., 2022; Ghosh et al., 2025; Barua et al., 2025; He et al., 2025). This motivates XCoT to go beyond prompt translation and exploit multilingual signals (Huang et al., 2023; Wang et al., 2025a; Chai et al., 2025; Ahuja et al., 2025). Recently, XCoT studies instruction tuning for CoT transfer, distilling reasoning from high-resource to low-resource languages (Upadhayay and Behzadan, 2023; Kuulmets et al., 2024; Chai et al., 2025; Weihua et al., 2025).

[7]The prompt is provided in Appendix D

### 4.2 Cross-lingual Chain-of-Thought Self-consistency

Self-consistency improves CoT by sampling multiple paths and selecting the most consistent answer (Wang et al., 2022; Aggarwal et al., 2023; Wang et al., 2025b).

Cross-lingual self-consistent prompting extends this to XCoT paths (Qin et al., 2023), while AUTOCAP automates language selection and learns language-specific weights (Zhang et al., 2024). Cross-lingual Tree-of-Thoughts performs multilingual search with branching reasoning and aggregation (Ranaldi et al., 2024). $L^2$ leverages multilingual unification learning and decoding-time interventions (Chen et al., 2025b). Best-of-L ranks multilingual candidates with a cross-lingual reward model for math reasoning (Rajaee et al., 2025) and Multidimensional Consistency aggregates signals across input perturbations to improve robustness (Lai et al., 2025).

Compared to these methods, our focus is efficiency-oriented allocation of test-time computation. We align intermediate states across languages in a unified logic space, which makes partial reasoning trajectories directly comparable during decoding and enables query-adaptive language selection together with online pruning.

## 5 Conclusion

We proposed UL-XCoT, an efficient cross-lingual reasoning framework that leverages a unified logic mechanism to better allocate compute resources. By selecting query-adaptive candidate languages and pruning inconsistent XCoT dynamically, UL-XCoT reduces redundancy at both the language and token levels while maintaining strong cross-lingual reasoning quality.

Experiments on PolyMath and MMLU-ProX-Lite show competitive accuracy with significantly higher efficiency and stable gains on a data-driven low-resource subset.

## Limitations

Prior interpretability work suggests that transformer hidden states encode information-rich representations that can be meaningfully inspected and analyzed (Yang et al., 2024; Ghandeharioun et al., 2024; Skean et al., 2025). Our method builds on this observation by leveraging hidden-state representations to compare cross-lingual understanding differences in a unified logic space and to monitor the quality of reasoning trajectories for online pruning. Therefore, UL-XCoT assumes white-box access to hidden states. Applicability to strict black-box LLM APIs remains to be validated.

## Acknowledgement

This work was supported by the National Natural Science Foundation of China (NSFC) via grants 92570120 and 62306342. This work was supported by the Scientific Research Fund of Hunan Provincial Education Department (24B0001). This work was sponsored by the Excellent Young Scientists Fund in Hunan Province (2024JJ4070), the Science and Technology Innovation Program of Hunan Province under Grant 2024RC3024. This study was also funded by the Open Project of the Text Computing and Cognitive Intelligence Ministry of Education Engineering Research Center (No. TCCI250101).

## References

Pranjal Aggarwal, Aman Madaan, Yiming Yang, et al. 2023. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with llms. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12375–12396.

Sanchit Ahuja, Praneetha Vaddamanu, and Barun Patra. 2025. Efficientxlang: Towards improving token efficiency through cross-lingual reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 15612–15624.

Sangmin Bae, Jongwoo Ko, Hwanjun Song, and Se-Young Yun. 2023. Fast and robust early-exiting framework for autoregressive language models with synchronized parallel decoding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5910–5924.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2024. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 749–775.

Josh Barua, Seun Eisape, Kayo Yin, and Alane Suhr. 2025. Long chain-of-thought reasoning across languages. *arXiv preprint arXiv:2508.14828*.

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

Kang Chen, Yaoning Wang, Kai Xiong, Zhuoka Feng, Wenhe Sun, Haotian Chen, and Yixin Cao. 2025a. Do llms signal when they're right? evidence from neuron agreement. *arXiv preprint arXiv:2510.26277*.

Kang Chen, Mengdi Zhang, and Yixin Cao. 2025b. Less data less tokens: Multilingual unification learning for efficient test-time reasoning in llms. *arXiv preprint arXiv:2506.18341*.

Nuo Chen, Zinan Zheng, Ning Wu, Ming Gong, Dongmei Zhang, and Jia Li. 2024a. Breaking language barriers in multilingual mathematical reasoning: Insights and observations. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7001–7016.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025c. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*.

Qiguang Chen, Libo Qin, Jiaqi Wang, Jinxuan Zhou, and Wanxiang Che. 2024b. Unlocking the capabilities of thought: A reasoning boundary framework to quantify and optimize chain-of-thought. *Advances in Neural Information Processing Systems*, 37:54872–54904.

Jiahai Feng, Stuart Russell, and Jacob Steinhardt. 2024. Monitoring latent world states in language models with propositional probes. *arXiv preprint arXiv:2406.19501*.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024. Patchscopes: A unifying framework for inspecting hidden representations of language models. *arXiv preprint arXiv:2401.06102*.

Akash Ghosh, Debayan Datta, Sriparna Saha, and Chirag Agarwal. 2025. A survey of multilingual reasoning in language models. *arXiv preprint arXiv:2502.09457*.

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

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*.

Yifei He, Alon Benhaim, Barun Patra, Praneetha Vaddamanu, Sanchit Ahuja, Parul Chopra, Vishrav Chaudhary, Han Zhao, and Xia Song. 2025. Scaling laws for multilingual language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 4257–4273.

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

Kaiyu Huang, Fengran Mo, Xinyu Zhang, Hongliang Li, You Li, Yuanchi Zhang, Weijian Yi, Yulong Mao, Jinchen Liu, Yuzhuang Xu, et al. 2026. A survey on large language models with multilingualism: Recent advances and new frontiers. *Artificial Intelligence Review*.

Ammar Khairi, Daniel D'souza, Ye Shen, Julia Kreutzer, and Sara Hooker. 2025. When life gives you samples: The benefits of scaling up inference compute for multilingual llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27547–27571.

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

Hele-Andra Kuulmets, Taido Purason, Agnes Luhtaru, and Mark Fishel. 2024. Teaching llama a new language through cross-lingual knowledge transfer. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3309–3325.

Huiyuan Lai, Xiaoyu Zhang, and Malvina Nissim. 2025. Multidimensional consistency improves reasoning in language models. *arXiv preprint arXiv:2503.02670*.

Viet Dac Lai, Nghia Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Huu Nguyen. 2023. Chatgpt beyond english: Towards a comprehensive evaluation of large language models in multilingual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13171–13189.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*.

OpenAI. 2024. Gpt-4o mini: advancing cost-efficient intelligence. OpenAI Blog. Accessed: 2026-01-05.

Libo Qin, Qiguang Chen, Xiachong Feng, Yang Wu, Yongheng Zhang, Yinghui Li, Min Li, Wanxiang Che, and Philip S Yu. 2026. Large language models meet nlp: A survey. *Frontiers of Computer Science*, 20(11):2011361.

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

Libo Qin, Qiguang Chen, Yuhang Zhou, Zhi Chen, Yinghui Li, Lizi Liao, Min Li, Wanxiang Che, and Philip S Yu. 2025. A survey of multilingual large language models. *Patterns*, 6(1).

Sara Rajaee, Rochelle Choenni, Ekaterina Shutova, et al. 2025. Best-of-l: Cross-lingual reward modeling for mathematical reasoning. *arXiv preprint arXiv:2509.15811*.

Leonardo Ranaldi, Giulia Pucci, Federico Ranaldi, Elena Sofia Ruzzetti, and Fabio Massimo Zanzotto. 2024. A tree-of-thoughts to broaden multi-step reasoning across languages. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1229–1241.

Matthew Renze and Erhan Guven. 2024. The benefits of a concise chain of thought on problem-solving in large language models. *arXiv preprint arXiv:2401.05618*.

Lucas Resck, Isabelle Augenstein, and Anna Korhonen. 2025. Explainability and interpretability of multilingual large language models: A survey. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20465–20497.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, et al. 2022. Language models are multilingual chain-of-thought reasoners. *arXiv preprint arXiv:2210.03057*.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. 2025. Layer by layer: Uncovering hidden representations in language models. *arXiv preprint arXiv:2502.02013*.

Khanh-Tung Tran, Barry O'Sullivan, and Hoang D Nguyen. 2025a. Scaling test-time compute for low-resource languages: Multilingual reasoning in llms. *arXiv e-prints*, pages arXiv–2504.

Khanh-Tung Tran, Nguyet-Hang Vu, Barry O'Sullivan, and Harry Nguyen. 2025b. Disentangling language understanding and reasoning structures in cross-lingual chain-of-thought prompting. In *EMNLP 2025*.

Bibek Upadhayay and Vahid Behzadan. 2023. Taco: Enhancing cross-lingual transfer for low-resource languages in llms through translation-assisted chain-of-thought processes. *arXiv preprint arXiv:2311.10797*.

Teng Wang, Zhenqi He, Wing-Yin Yu, Xiaojin Fu, and Xiongwei Han. 2025a. Large language models are good multi-lingual learners: When llms meet cross-lingual prompts. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4442–4456.

Weiqin Wang, Yile Wang, and Hui Huang. 2025b. Ranked voting based self-consistency of large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14410–14426.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yiming Wang, Pei Zhang, Siyuan Huang, Baosong Yang, Zhuosheng Zhang, Fei Huang, and Rui Wang. 2025c. Sampling-efficient test-time scaling: Self-estimating the best-of-n sampling in early decoding. *arXiv preprint arXiv:2503.01422*.

Yiming Wang, Pei Zhang, Jialong Tang, Haoran Wei, Baosong Yang, Rui Wang, Chenshu Sun, Feitong Sun, Jiran Zhang, Junxuan Wu, et al. 2025d. Polymath: Evaluating mathematical reasoning in multilingual contexts. *arXiv preprint arXiv:2504.18428*.

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

Zheng Weihua, Roy Ka-Wei Lee, Zhengyuan Liu, Wu Kui, Aiti Aw, and Bowei Zou. 2025. Ccl-xcot: An efficient cross-lingual knowledge transfer method for mitigating hallucination generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 1768–1788.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. 2025. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*.

Weihao Xuan, Rui Yang, Heli Qi, Qingcheng Zeng, Yunze Xiao, Aosong Feng, Dairui Liu, Yun Xing, Junjue Wang, Fan Gao, Jinghui Lu, Yuang Jiang, Huitao Li, Xin Li, Kunyu Yu, Ruihai Dong, Shangding Gu, Yuekang Li, Xiaofei Xie, Felix Juefei-Xu, Foutse Khomh, Osamu Yoshie, Qingyu Chen, Douglas Teodoro, Nan Liu, Randy Goebel, Lei Ma, Edison Marrese-Taylor, Shijian Lu, Yusuke Iwasawa, Yutaka Matsuo, and Irene Li. 2025. MMLU-ProX: A multilingual benchmark for advanced large language model evaluation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1513–1532, Suzhou, China. Association for Computational Linguistics.

Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. 2024. Regularizing hidden states enables learning generalizable reward model for llms. *Advances in Neural Information Processing Systems*, 37:62279–62309.

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. 2025a. Reasoning models know when they're right: Probing hidden states for self-verification. *arXiv preprint arXiv:2504.05419*.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, et al. 2025b. A survey on test-time scaling in large language models: What, how, where, and how well? *arXiv preprint arXiv:2503.24235*.

Yongheng Zhang, Qiguang Chen, Min Li, Wanxiang Che, and Libo Qin. 2024. Autocap: Towards automatic cross-lingual alignment planning for zero-shot chain-of-thought. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9191–9200.

Weixiang Zhao, Jiahe Guo, Yang Deng, Tongtong Wu, Wenxuan Zhang, Yulin Hu, Xingyu Sui, Yanyan Zhao, Wanxiang Che, Bing Qin, et al. 2025. When less language is more: Language-reasoning disentanglement makes llms better multilingual reasoners. *arXiv preprint arXiv:2505.15257*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

## A Mathematical Details of DCP

**Logic-space curvature signal.** Building upon hidden-state based self-truncation ideas (Wang et al., 2025c; Zhang et al., 2025a; Bae et al., 2023; Chen et al., 2025a), we quantify the within-model stability of a trajectory by measuring the curvature of projected hidden states across layers at each decoding step. After warm-up, at decoding moment $t$, for each active language path $\ell$, let $x_\ell^t$ denote the current prefix. Let $m \in \mathcal{M} = \{m_s, \ldots, m_e\}$ be the monitored Transformer layers. We define the position-averaged projected hidden state:

$$h_{t,m}^{\ell} := \hat{H}_m^t(x_\ell).$$

$$\hat{H}_m^t(x, \ell) = \frac{1}{|x_\ell^t|} \sum_{i=1}^{|x_\ell^t|} \tilde{H}_{i,m}^t(x, \ell).$$

where $\tilde{H}_{i,m}^t(x, \ell)$ is the projected hidden state at token position $i$ and layer $m$.

We quantify layer-to-layer local changes in magnitude and direction. Define cosine similarity:

$$\cos(u, v) := \frac{u^\top v}{\|u\| \, \|v\|}.$$

For each adjacent layer pair $(m-1, m)$, define the magnitude change:

$$\delta_M^{(t,m)}(\ell) := \left\| h_{t,m}^{\ell} - h_{t,m-1}^{\ell} \right\|_2.$$

Define the angular change:

$$\delta_A^{(t,m)}(\ell) := \arccos\big(\cos(h_{t,m}^{\ell}, h_{t,m-1}^{\ell})\big).$$

We normalize by the end-to-end (chord) change across layers at the same step $t$. Define:

$$\Delta_M^t(\ell) := \left\| h_{t,m_e}^{\ell} - h_{t,m_s}^{\ell} \right\|_2.$$

$$\Delta_A^t(\ell) := \arccos\big(\cos(h_{t,m_e}^{\ell}, h_{t,m_s}^{\ell})\big).$$

The layer-wise curvature ratios are:

$$r_M^t(\ell) := \frac{\sum_{m=m_s+1}^{m_e} \delta_M^{(t,m)}(\ell)}{\Delta_M^t(\ell)}.$$

$$r_A^t(\ell) := \frac{\sum_{m=m_s+1}^{m_e} \delta_A^{(t,m)}(\ell)}{\Delta_A^t(\ell)}.$$

Finally, we define a **Logic-space curvature signal** to measure the evolution of a path:

$$\kappa^t(\ell) := r_M^t(\ell) - r_A^t(\ell).$$

**Divergence test.** To avoid false positives caused by a global drift shared by all paths, we implement an indicator $\mathbb{I}_\kappa t$ using both absolute and relative pairwise spread. Specifically, define the maximum absolute spread

$$\Delta_{\max}^t := \max_{\ell \neq \ell'} \left| \kappa^t(\ell) - \kappa^t(\ell') \right|,$$

the maximum relative spread

$$R_{\max}^t := \max_{\ell \neq \ell'} \frac{\left| \kappa^t(\ell) - \kappa^t(\ell') \right|}{\max\big(|\kappa^t(\ell)|, \ |\kappa^t(\ell')|, \ \delta\big)},$$

and the mean relative spread over all unordered pairs can be expressed as

$$R_{\text{mean}}^t := \mathbb{E}_{\ell \neq \ell'} \left[ \frac{\left| \kappa^t(\ell) - \kappa^t(\ell') \right|}{\max\big(|\kappa^t(\ell)|, \ |\kappa^t(\ell')|, \ \delta\big)} \right],$$

where $\delta > 0$ is a small constant for numerical stability. We declare step $t$ as **divergent** iff all three conditions hold:

$$\mathbb{I}_\kappa^t = \mathbb{I}\big[\Delta_{\max}^t > \varepsilon_{\text{abs}}\big] \cdot \mathbb{I}\big[R_{\max}^t > \varepsilon_{\text{rel}}\big] \cdot \mathbb{I}\big[R_{\max}^t \geq \gamma \, R_{\text{mean}}^t\big].$$

**Divergence detection and scoring.** Let $S_t^*$ denote the set of active paths at decoding step $t$ and $N_t = |S_t|$. For each path $\ell \in S_t$, let $\kappa^t(\ell)$ be its divergence descriptor at step $t$. We start monitoring after a warm-up period and define the first post-warm-up divergence step as

$$c := \min\{\, t \geq T_{\text{warm}} \mid \mathbb{I}_\kappa^t = 1 \,\}.$$

We only score paths within a fixed window

$$t \in [c, \, c + \tau].$$

**Per-step point assignment.** We set $K_t' := \min(K', N_t)$ and assign each path $\ell \in S_t$ a binary point $\texttt{score}(S_t | x_\ell, \ell') \in \{0, 1\}$:

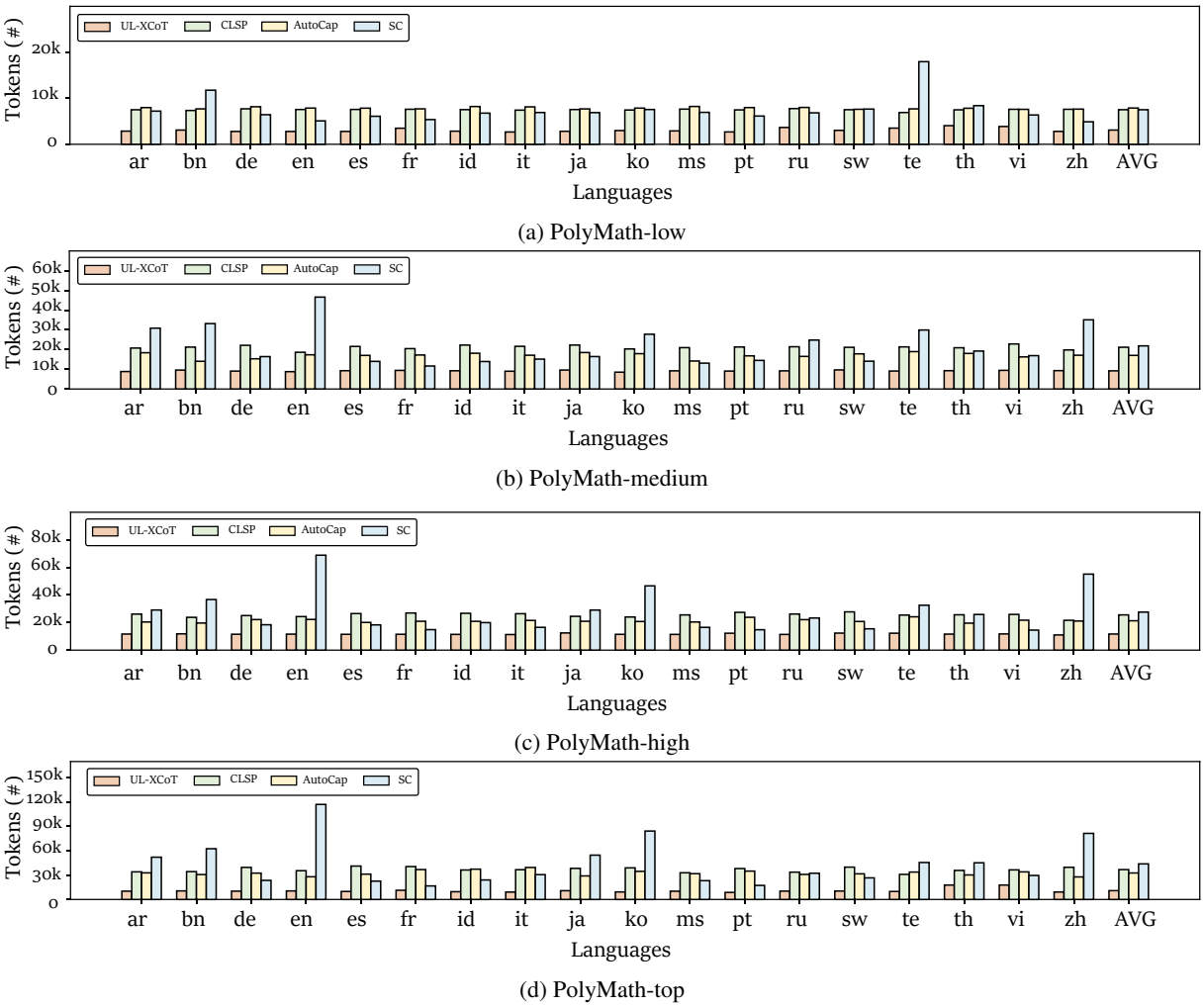
$$\begin{cases} \mathbb{I}[\ell \in R_t], & \mathbb{I}_\kappa^t = 0 \ , \\ \mathbb{I}[\ell \in W_t], & \mathbb{I}_\kappa^t = 1 \ , \end{cases} \tag{12}$$

where, if step $t$ is non-divergent, we sample $R_t \subseteq S_t$ uniformly at random without replacement with $|R_t| = K'$. If step $t$ is divergent, we keep the $K'$ most central paths by minimizing the average distance to the cohort in the divergence space:

$$g^t(\ell) := \frac{1}{\max(1, N_t - 1)} \sum_{\ell' \in S_t \setminus \{\ell\}} \left| \kappa^t(\ell) - \kappa^t(\ell') \right|$$

$$W_t := \text{TopK}'_{\ell \in S_t}\big(-g^t(\ell)\big).$$

Equivalently, $W_t$ contains the $K_t'$ paths with the smallest $g^t(\ell)$, with ties broken arbitrarily.




(a) PolyMath-low

(b) PolyMath-medium

(c) PolyMath-high

(d) PolyMath-top

Figure 11: Decoding token cost during generation across PolyMath difficulty levels.

## B Detailed Efficiency Analysis

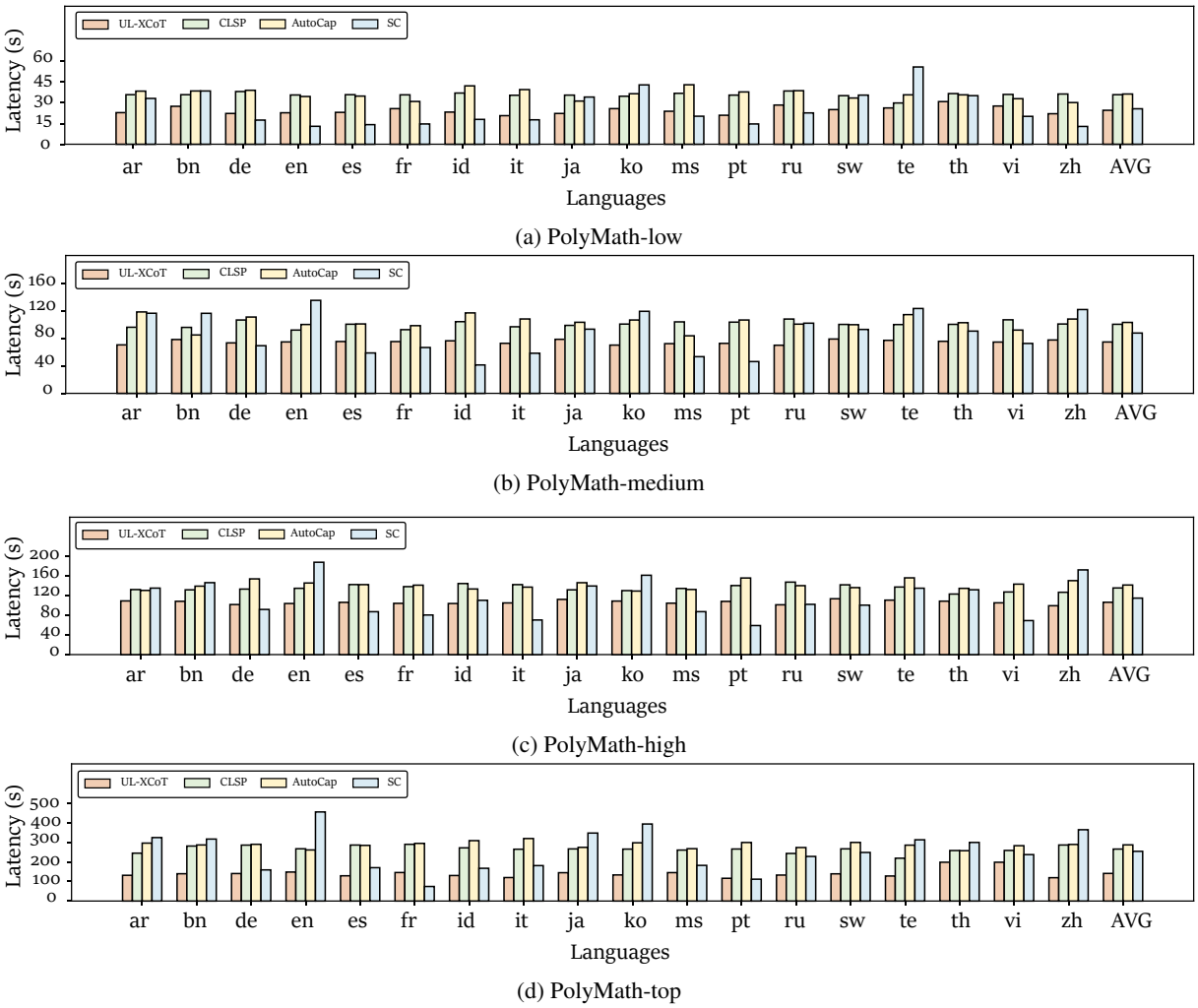
This appendix analyzes the efficiency results in Sec. 3.2.1 by PolyMath difficulty level, providing additional evidence for the less languages, less tokens motivation.

### B.1 Token cost across difficulty levels.

Figure 11 reports the decoding token cost over four PolyMath difficulty levels, broken down by language. As difficulty increases from low to top, all methods exhibit higher token consumption, indicating longer reasoning traces on harder problems. Across all languages and levels, UL-XCoT consistently incurs fewer tokens than the baselines, showing that query-adaptive language selection and online trajectory pruning can reduce unnecessary generation without relying on extra sampling. The gap becomes more evident on harder subsets (high-/top), where baseline decoding tends to produce longer and more variable traces, while UL-XCoT keeps token usage more stable and cost-efficient.

### B.2 Wall-clock latency across difficulty levels.

Figure 12 reports decoding wall-clock latency during generation across four PolyMath difficulty levels, broken down by language. Latency increases monotonically from low to top for all methods, reflecting the longer and more compute-intensive reasoning required by harder problems. Across nearly all languages and all levels, UL-XCoT consistently achieves the lowest latency and the best AVG, indicating a more efficient decoding profile than CLSP, AUTOCAP, and SC. Notably, the advantage becomes more pronounced on high/top, where baseline methods exhibit larger variance and occasional long-tail spikes, while UL-XCoT remains more stable. This improvement is attributed to selecting candidate languages before decoding and dynamically pruning low-quality XCoT paths early, thereby avoiding unnecessary generation early and reducing end-to-end serving time.




(a) PolyMath-low

(b) PolyMath-medium

(c) PolyMath-high

(d) PolyMath-top

Figure 12: Decoding wall-clock latency during generation across PolyMath difficulty levels.

Table 4: Difficulty considered performance on PolyMath across 18 languages and four difficulty levels.

| DW-ACC | ar | bn | de | en | es | fr | id | it | ja | ko | ms | pt | ru | sw | te | th | vi | zh | AVG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CoT (Wei et al., 2022) | 10.4 | 7.1 | 12.8 | 11.8 | 14.1 | 13.0 | 10.9 | 11.0 | 11.3 | 9.3 | 9.3 | 13.3 | 12.3 | 2.1 | 2.8 | 10.6 | 11.4 | 14.0 | 10.4 |
| CLP (Qin et al., 2023) | 9.0 | 7.9 | 11.9 | 13.4 | 10.3 | 13.3 | 9.9 | 11.6 | 13.8 | 10.7 | 12.5 | 11.1 | 12.1 | 11.0 | 9.2 | 8.1 | 11.1 | 11.4 | 11.0 |
| SC (Wang et al., 2022) | **18.9** | 12.5 | 14.6 | **20.2** | 16.5 | 16.7 | 14.7 | 17.6 | 11.4 | 19.4 | 14.4 | 15.5 | 17.4 | 4.7 | 6.2 | 14.5 | 15.5 | **22.6** | 15.2 |
| CLSP (Qin et al., 2023) | 18.2 | 16.9 | **20.8** | 19.4 | 16.8 | 18.8 | **20.7** | 16.7 | **19.7** | **20.5** | 18.5 | 16.7 | 17.7 | 19.2 | **18.8** | 18.5 | 18.6 | 17.7 | 18.6 |
| AUTOCAP (Zhang et al., 2024) | 18.3 | 18.3 | 17.8 | 17.3 | 18.3 | **19.4** | 18.8 | 16.1 | 17.9 | 20.0 | **20.5** | **18.3** | 16.3 | 17.1 | 17.9 | 15.0 | 17.7 | 18.2 | 18.2 |
| ST-BoN (Wang et al., 2025c) | 13.8 | 12.4 | 11.7 | 12.7 | 12.2 | 14.2 | 13.4 | 14.2 | 13.9 | 13.4 | 14.7 | 14.1 | 12.6 | 12.0 | 12.3 | 15.0 | 14.3 | 13.1 | 13.3 |
| **UL-XCoT** | 18.5 | **19.1** | 19.3 | 19.7 | **19.1** | 19.1 | 20.0 | **18.1** | 17.8 | 17.3 | 19.3 | 17.4 | **20.1** | **21.8** | 18.0 | **20.0** | **19.7** | 18.0 | **19.0** |

## C Overall Performance on PolyMath

Table 4 provides the **PolyMath-Full** summary in terms of **DW-ACC**, which is the primary effectiveness metric used in our evaluation protocol. Different from Table 1 that reports accuracy separately for each difficulty level, DW-ACC **aggregates** performance over the four levels into a single difficulty-aware score, enabling a compact comparison of overall effectiveness across languages.

Overall, UL-XCoT achieves the best AVG DW-ACC, and remains consistently strong across the 18-language suite. This result directly supports that our efficiency-oriented framework substantially reduces inference cost, it stays competitive in difficulty-weighted accuracy on PolyMath-Full. In particular, the gains are not confined to a single high-resource language, but are distributed across languages, suggesting that the proposed CLS and DCP strategy guarantees cross-lingual reasoning quality under the same decoding budget.

# D Prompt Templates

## D.1 Concise Reasoning Prompt

We faithfully implement the reasoning setting using the following prompt, which enforces a concise thinking process and enables a clearer analysis of performance under such a regime (Renze and Guven, 2024; Xu et al., 2025).

Specifically, it (i) fixes the reasoning language to `<LANG_NAME>` to avoid cross-lingual leakage, (ii) caps the reasoning to at most `<STEP_NUM>` numbered steps to control verbosity and token budget, and (iii) restricts the final output to a single boxed answer outside the `<think>` block, ensuring a clean separation between intermediate reasoning and the model's final response. This standardized format makes results comparable and isolates efficiency gains attributable to the decoding strategy rather than prompt-induced length differences.

## D.2 Quality Judge Prompt

We use an **LLM-as-a-judge** prompt to score each candidate trajectory in a structured and machine-readable manner (Zheng et al., 2023; Li et al., 2024). We use GPT–4o mini as the LLM in our experiments (OpenAI, 2024). The judge takes the `question`, `reference_answer`, `candidate_answer`, and the candidate CoT as input, and is constrained to output **only** a JSON object matching a fixed schema. It assigns integer scores in $[0, 100]$ on six dimensions: correctness (exact match to the reference answer), step validity (logical soundness without jumps), faithfulness (no hallucinated facts), completeness (covers key constraints), conciseness (non-redundant), and compliance (language constraints). The final `overall` score is a weighted average with the largest weight on correctness.

**concise-reasoning template**

You are an expert in mathematical / geometric reasoning.

Think strictly in <LANG_NAME> step by step. Do not use any other language.

**Format:**

`<think>`

Step 1: ...

Step 2: ...

...

Step N: ...

`</think>`

`$\boxed {FINAL\_ANSWER}$`

**Hard Rules:**

1) All intermediate reasoning MUST be inside a single `<think>...</think>` block, written only in <LANG_NAME>.

2) At most <STEP_NUM> numbered steps. Be concise and avoid repetition.

3) Outside `</think>` you may output ONE line only: `$\boxed {FINAL\_ANSWER}$`.

4) Do NOT restate the problem. Do NOT add any explanation, comments, or extra text after the boxed answer.

5) If the result is an expression, keep it simplified. If numeric, give an exact value when possible.

**Question:**

<QUERY>

**Nnotes:**

- Use standard math notation. Keep symbols/variables as-is.

- If you reach a conclusion early, stop immediately and output the boxed answer.

**quality-judge-prompt**

You are a strict grader. Output ONLY JSON matching the schema.

Score scale (IMPORTANT):

- All scores are INTEGERS from 0 to 100 (100 is best).
- 0–20: very poor, 40–60: mediocre, 70–80: good, 85–95: very good, 96–100: near-perfect.
- DO NOT use a 1–5 scale.

**Question:**

<question>

**Reference Answer:**

<reference_answer>

**Candidate Answer:**

<candidate_answer>

**Candidate Reasoning / CoT:**

<candidate_cot>

**Dimension rules:**

- correctness: 100 if candidate answer matches reference answer (allow trivial formatting), else 0.
- step_validity: penalize jumps/invalid inference; 100 means each step is logically justified.
- faithfulness: penalize invented facts not in question or derivable; 100 means fully grounded.
- completeness: 100 means all key constraints/calculations covered.
- conciseness: 100 means no redundancy; lower if repetitive.
- compliance: 100 means follows required format/language constraints.

overall: weighted average (correctness has the largest weight).