

Multilingual Chain of Thought Compression via Cross-lingual Entropy Gap

Jiarui Wan^{1,2}, Yufeng Chen^{1,2}, Songming Zhang^{1,2}, Rui Qi^{1,2},
Xue Zhang^{1,2}, Jinan Xu^{1,2}, Kaiyu Huang^{1,2,*}

¹ Key Laboratory of Big Data & Artificial Intelligence in Transportation
(Beijing Jiaotong University), Ministry of Education

² School of Computer Science and Technology, Beijing Jiaotong University
{24120380}@bjtu.edu.cn

Abstract

Current large reasoning models (LRMs) prefer to generate verbose chain-of-thought (CoT) traces (a.k.a. the “overthinking” issue), especially in multilingual scenarios due to the imbalanced proficiency across different languages. However, existing studies on CoT compression focus exclusively on English and overlook language characteristics, resulting in performance degradation in multilingual scenarios. To address this issue, we develop a multilingual CoT compression method based on reinforcement learning, named **Language-aware Entropy-based (LAE)**, which optimizes LLMs with multilingual group sampling and entropy gap-based rewards. Specifically, we extend GRPO with multilingual group sampling, where each query is jointly optimized across multiple languages. We further design a language-aware entropy-gap reward that penalizes redundant uncertainty in the thinking phase and aligns reasoning complexity across languages, enabling concise yet accurate multilingual CoT reasoning. Extensive experiments on the multilingual reasoning benchmarks MGSM, MMATH and MMMLU demonstrate that our method achieves competitive performance and generalization, compared with several strong baselines.

1 Introduction

Large language models (LLMs) have demonstrated strong capabilities in complex reasoning tasks (Jaech et al., 2024; Guo et al., 2025; Yang et al., 2025a; Team et al., 2025) when equipped with chain-of-thought (CoT) prompting, which encourages models to generate intermediate reasoning steps before producing a final answer. Despite its effectiveness, CoT reasoning often produces excessively long reasoning traces, resulting in high computational costs and latency during inference (Wu et al., 2025; Kumar et al., 2025). Consequently,

enhancing reasoning efficiency has emerged as a critical research direction for the scalable deployment of reasoning-capable LLM, particularly through the compression of redundant reasoning steps.

Recent studies have explored CoT compression techniques that aim to reduce the number of reasoning tokens while preserving or even improving reasoning accuracy. Existing approaches typically rely on signals such as token importance scores (Xia et al., 2025) and confidence estimates (Yang et al., 2025b) derived from the reasoning process to prune redundant reasoning tokens, aiming to alleviate redundancy of the reasoning path. These methods have shown promising results on English reasoning benchmarks such as GSM8K and MATH, suggesting that LLMs can learn to produce shorter yet accurate reasoning processes. These CoT compression methods primarily focus on monolingual settings, especially English, largely because existing methods treat reasoning compression as a language-agnostic optimization problem, ignoring cross-lingual variations in reasoning dynamics (Xia et al., 2025; Kang et al., 2025; Yuan et al., 2025; Li et al., 2025a). As a result, these methods fail to generalize effectively to multilingual or cross-lingual settings.

As shown in Table 1, directly applying existing CoT compression methods to multilingual settings exposes two key failure modes. First, performance drops significantly in non-English languages. For instance, applying Step Entropy (Li et al., 2025b) to Bengali reduces accuracy by 15.6 points. This degradation stems from the fact that compression signals, such as entropy or confidence scores calibrated on English, cannot reliably distinguish redundant tokens from essential reasoning steps in lower-resource languages. Due to inherently higher token-level uncertainty, these methods tend to over-prune necessary reasoning, thereby degrading performance. Second, some languages

*Kaiyu Huang is the corresponding author.

Language	Metric	Original	Compressed CoT
EN	Acc.	84.8	87.12 $\uparrow$ (+2.32)
	Tok.	518	422 $\uparrow$ (+0.19)
SW	Acc.	7.44	6.24 $\downarrow$ (-1.20)
	Tok.	1032	1507 $\downarrow$ (-0.46)
BN	Acc.	56.48	40.88 $\downarrow$ (-15.6)
	Tok.	1109	572 $\uparrow$ (+0.48)

Table 1: Performance of the compression method (Li et al., 2025b) based on R1-Distill-Qwen-7B across languages, evaluated on the MGSM (Shi et al., 2022) dataset. “Acc.” denotes the accuracy and “Tok.” denotes the average generated response token numbers.

exhibit token expansion rather than compression. In Swahili, token counts increase from 1,032 to 1,507 after compression, a 46% inflation. This occurs because English-centric compression signals are misaligned with the structural properties of typologically distant languages. Erroneous pruning of critical reasoning steps forces the model into repeated re-exploration and self-verification, ultimately producing longer outputs. Overall, without modeling language-specific uncertainty, existing methods fail to generalize and may even exacerbate verbosity in multilingual settings.

To address these limitations, we propose a **Language-aware Entropy-based (LAE)** reinforcement learning method for multilingual CoT compression. The core idea is to exploit cross-lingual differences in uncertainty to identify redundant reasoning and encourage concise reasoning chains. A central design is multilingual group sampling, which extends the group mechanism in GRPO by constructing groups that contain one reasoning trajectory per language for the same underlying query. This design is motivated by the observation that, for identical problems, reasoning length and uncertainty vary substantially across languages due to differences in language ability, tokenization density, and syntactic structure. As a result, standard single-language group sampling, which relies on multiple stochastic rollouts within one language, cannot disentangle necessary reasoning complexity from language-induced verbosity.

By contrast, multilingual groups enable direct comparison of reasoning trajectories across languages within the same optimization step, providing language-aware signals for compression. Building on this, we introduce a multilingual relative entropy-based reward that explicitly models cross-lingual uncertainty. Within each group, we identify

the trajectory with the most stable entropy profile as a reference, which serves as an anchor for cross-lingual knowledge transfer. We then impose relative redundancy penalties on more verbose trajectories, encouraging the model to suppress unnecessary reasoning steps and align reasoning behaviors across languages toward more concise and certain solutions. In this way, LAE grounds compression decisions in relative, language-aware evidence rather than language-agnostic thresholds, leading to more robust and effective multilingual reasoning compression. Experimental results show that LAE achieves 15% to 43% CoT compression across two datasets and three backbone models, while consistently improving reasoning accuracy.

Our contributions are summarized as follows:

- We propose a language-aware entropy-based method for multilingual CoT compression, which achieves efficient reasoning while improving performance.
- For multilingual settings, we propose multilingual group sampling and a language-aware reward function that explicitly accounts for language diversity within groups.
- We demonstrate that the proposed method generalizes competitively under limited data and language coverage across various LLMs, and that entropy serves as a natural uncertainty measure whose reduction suppresses overthinking while preserving essential reasoning information.

2 Related Work

CoT reasoning (Wei et al., 2022; Kojima et al., 2022; Wang et al., 2022) and large reasoning models (LRMs) (Jaech et al., 2024; Guo et al., 2025; Yang et al., 2025a; Team et al., 2025) have been shown to substantially improve performance via thinking in inference. However, explicit CoT reasoning incurs substantial computational and memory overhead, especially in long-context scenarios, and may expose sensitive reasoning traces (Team et al., 2025). Thus, due to the inefficiency caused by verbose intermediate steps, CoT compression aims to reduce reasoning length while preserving correctness and interpretability (Chen et al., 2024).

A branch of works compresses CoT by reducing the number or granularity of intermediate reasoning steps, for example early stopping based on hidden states during model inference (Yang et al., 2025b;

Qiao et al., 2025; Zhu et al., 2025) or removing redundant sub-derivations (Xia et al., 2025; Huang et al., 2025c; Zhao et al., 2025). Another branch of works explicitly constrains the length of generated CoT, either through constructing short CoT training data (Kang et al., 2025; Yuan et al., 2025; Yan et al., 2025) or training objectives that penalize overly long rationales (Li et al., 2025c).

To ensure an appropriate balance between efficiency and effectiveness, the compressed portions of the reasoning process should be genuinely redundant. In particular, Zeng et al. (2025) observe a systematic correlation between the token-level entropy changes exhibited by the model during the generation process and the evolution of its internal inference steps. Building on prior analyses of token-level entropy, Li et al. (2025b) further introduce step entropy to measure the information contribution of individual reasoning steps in CoT trajectories, enabling selective pruning of low-entropy steps. Moreover, Huang et al. (2025b) design a reward function based on the observation that model entropy varies across different stages of reasoning, guiding the model to reduce token generation during training, dynamically modulating reasoning depth or length according to the difficulty of the input (Jiang et al., 2025; Luo et al., 2025a).

While significant progress has been made in compressing CoT reasoning for monolingual settings, current research remains largely confined to English-centric reasoning spaces (Huang et al., 2025a, 2026; Zhang et al., 2026). In multilingual LLMs, however, disparities in syntax, morphology, and tokenization result in significant variations in structural patterns and information density across languages (Guan et al., 2025; Yang et al., 2026; Zhang et al., 2025b). These cross-lingual differences often undermine the effectiveness of traditional compression strategies, which are underexplored in the literature (Zhang et al., 2025a). Our method explicitly accounts for these variations through an adaptive, model-driven framework, bridging a critical gap in multilingual reasoning optimization.

3 Method

We present a novel approach to refining language model reasoning behavior in multilingual settings by extending the GRPO framework with a language-aware entropy-based mechanism. Our method leverages an adaptive selection of “Anchor

Language” in each multilingual group to promote concise and stable inference across languages. We first review the functionality of GRPO in multilingual CoT settings, then design a new group sampling strategy on top of the observation. Finally, we define the entropy gap-based reward function that explicitly penalizes redundant reasoning while preserving necessary uncertainty.

3.1 Preliminary

Unlike Proximal Policy Optimization (PPO) (Schulman et al., 2017), GRPO (Shao et al., 2024) estimates the baseline directly from the outputs of the reward model, thereby eliminating the need to train an explicit and potentially complex value function. Specifically, for each input query q , GRPO first sample a group of G candidate outputs $\{o_1, o_2, \dots, o_G\}$ from the old policy $\pi_{\theta_{\text{old}}}$ and compute corresponding reward $\{r_1, r_2, \dots, r_G\}$ by using reward function. Then, GRPO calculate the average reward and standard deviation for all responses within the group. For each response within the group, its advantage is defined as its normalized reward:

$$\hat{A}_{i,t} = \frac{r_i - \text{mean}(\{r_j\}_{j=1}^G)}{\text{std}(\{r_j\}_{j=1}^G)}. \quad (1)$$

This normalization scheme emphasizes intra-group differences among model responses to the same query. In section 3.2, we introduce our modifications, which further exploit these intra-group differences to a greater extent. Finally, GRPO uses advantage to update model parameters.

3.2 Multilingual Group Sampling

The binary reward function adopted by GRPO is simple yet effective in practice. However, its supervision signal depends solely on the correctness of the final answer and fails to exploit the structural information available during sampling. As mentioned in preceding Section 1, we observe that for the same query, the lengths of the generated CoT vary substantially across different input languages. These differences are often not driven by the intrinsic reasoning requirements of the problem, but instead arise from uncertainty induced by disparities in language resource availability and expression patterns. Consequently, a correctness-only reward is insufficient to constrain excessive exploratory reasoning in multilingual settings, particularly for low-resource languages.

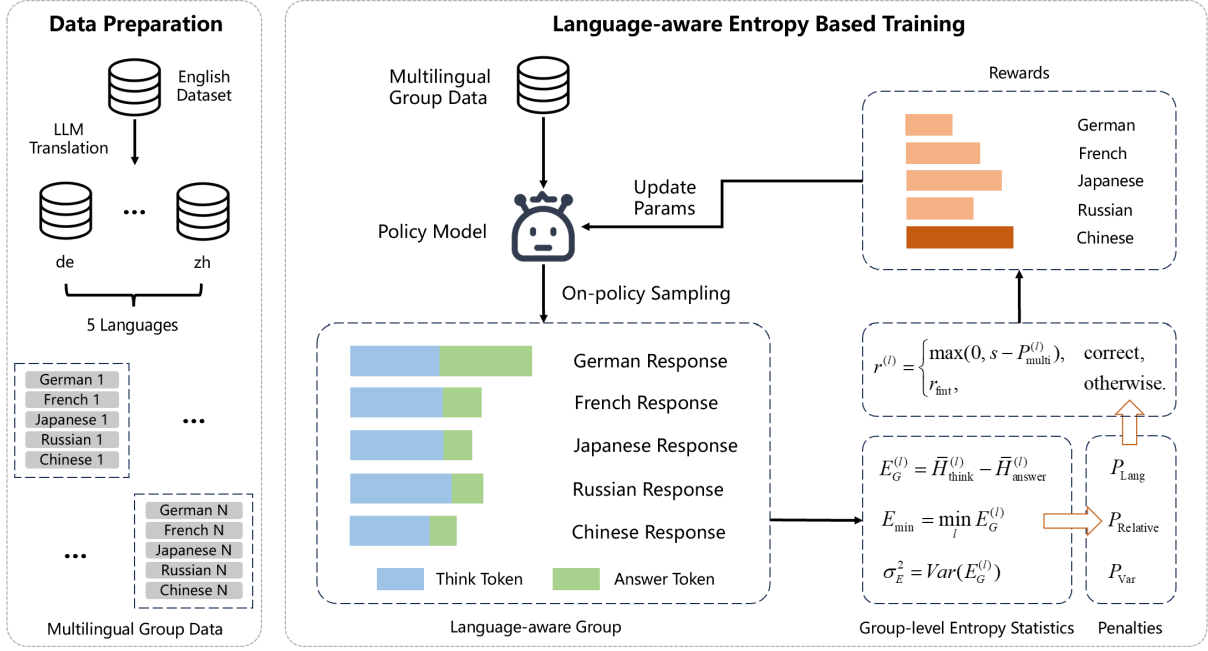


Figure 1: Overview of our method. Each query is translated into multiple languages to form multilingual groups. The policy model generates one response per language using on-policy sampling (Language-aware Group). The entropy gap is computed between thinking and answer phases ($E_G^{(l)}$), and the group-level statistics are derived to construct language-aware redundancy penalties ($E_{\min}$ and σ_E^2). These penalties are combined with answer correctness to form the final reward ($P_{\text{Lang}} + P_{\text{Relative}} + P_{\text{Var}}$), guiding the model to learn concise and aligned reasoning across languages.

To explicitly model cross-lingual differences in reasoning behavior, we modify the group sampling strategy used in GRPO. As shown in Figure 1, instead of performing multiple stochastic samples for the same problem instance, we construct each sampling group from different language versions of the same problem. Specifically, for each underlying problem x , we construct a group by translating the prompt into L different languages $\{x^{(1)}, x^{(2)}, \dots, x^{(L)}\}$. The policy then generates one response $y^{(l)} \sim \pi_{\theta}(\cdot | x^{(l)})$ per language, forming a multilingual group $G(y) = \{y^{(1)}, y^{(2)}, \dots, y^{(L)}\}$ of size L . In this way, each group naturally corresponds to a multilingual reasoning alignment problem, enabling the model to directly compare reasoning behaviors across languages within the group. We provide a comparison of the computational overhead between multilingual group sampling and GRPO in Appendix H.

3.3 Entropy Gap Modeling

To quantify the determinacy of reasoning processes across different languages, a natural signal of redundancy is the model’s uncertainty, as excessive exploration in reasoning is typically reflected by elevated entropy.

A well-calibrated reasoning process should gradually reduce uncertainty from exploration (thinking phase) to exploitation (answer phase). Excessive entropy in the thinking phase relative to the answer phase therefore indicates unnecessary exploratory reasoning. An illustrative example of this phenomenon is provided in Appendix D. Therefore, we define redundancy as the entropy gap between the thinking phase and the answer phase. Concretely, for each language l , we compute the average entropy during the thinking phase $H_{\text{think}}^{(l)}$ and the average entropy during the answer phase $H_{\text{answer}}^{(l)}$. Given a sampled sentence consisting of N tokens $y^{(l)} = (y_1, y_2, \dots, y_N)$, token level entropy are computed using old policy $\pi_{\theta_{\text{old}}}$:

$$H_i = - \sum_{v \in \mathcal{V}} \pi_{\theta_{\text{old}}}(v | y_{<i}) \log \pi_{\theta_{\text{old}}}(v | y_{<i}), \quad (2)$$

where $i = 1, \dots, N$.

Let t be the index of the special token </think> in the sequence. Thinking entropy and answer entropy are defined as the average token-level entropy, computed by separating the model output at the special token </think> :

$$\bar{H}_{\text{think}} = \frac{1}{t-1} \sum_{i=1}^{t-1} H_i, \quad (3)$$

$$\bar{H}_{\text{answer}} = \frac{1}{N-t} \sum_{i=t+1}^N H_i, \quad (4)$$

and then the **entropy gap** as:

$$E_G^{(l)} = H_{\text{think}}^{(l)} - H_{\text{answer}}^{(l)}. \quad (5)$$

This design ensures that the penalty is applied only when uncertainty in the thinking phase is disproportionately large relative to the natural uncertainty required by the final answer. Importantly, answer entropy serves as a stable reference rather than an optimization target, preventing the model from indiscriminately suppressing entropy across all generation stages. Using an entropy gap further normalizes language-specific entropy scales, mitigating unfair comparisons across languages within the same multilingual group.

3.4 Language-Aware Entropy-Based Reward

Based on the above analysis, we propose a multilingual redundancy reward to constrain redundant behavior in multilingual reasoning. Within a multilingual group, we first compute group minimum entropy gap $E_{\min}$ and group entropy gap variance σ_E^2 :

$$E_{\min} = \min_l E_G^{(l)}, \quad (6)$$

$$\bar{E} = \frac{1}{L} \sum_l E_G^{(l)}, \quad (7)$$

$$\sigma_E^2 = \frac{1}{L} \sum_l \left(E_G^{(l)} - \bar{E} \right)^2, \quad (8)$$

where L is the number of languages within the sampling group and $l \in \{\text{de, fr, ja, ru, zh}\}$. Regarding the selection of languages, we adopt a prior knowledge and Huang et al. (2025d) that is provided in Appendix B.1.

After calculating group-level statistics, We then compute the reward which consists of three components:

In-language Redundancy. In-language redundancy captures excessive length or unnecessary complexity within the reasoning process of a single language. When a language exhibits an overly long reasoning path, its entropy gap tends to be high, indicating excessive uncertainty. We penalize this

surplus uncertainty to encourage concise and efficient reasoning within each language. This penalty is expressed by the formula as:

$$P_{\text{Lang}} = \max(0, E_G^{(l)}). \quad (9)$$

Relative Redundancy. Within a multilingual group, different languages may exhibit varying levels of reasoning complexity. We select the language with the minimum entropy gap as the reference language, corresponding to the most concise reasoning path for the given problem. Additional penalties are applied only when a language’s entropy gap significantly exceeds this minimum. This design avoids forcing uniformly short reasoning across all languages and instead aligns each language toward the most concise reasoning path observed within the group. This penalty is expressed by the formula:

$$P_{\text{Relative}} = E_G^{(l)} - E_{\min}. \quad (10)$$

Group Variance Penalty. The group variance measures inconsistency in reasoning complexity across languages. Large variance indicates that the model adopts substantially different reasoning strategies in multilingual settings. By penalizing this variance, we encourage the model to learn more consistent cross-lingual reasoning behaviors, maintaining similar reasoning structures across languages and thereby improving overall stability and generalization. This penalty is expressed by the formula:

$$P_{\text{var}} = \sigma_E^2. \quad (11)$$

Finally, we add these three parts together:

$$P_{\text{multi}}^{(l)} = P_{\text{Lang}} + P_{\text{Relative}} + P_{\text{var}}. \quad (12)$$

For samples with correct answers, we subtract the three redundancy penalties, weighted by a coefficient λ , from a base correctness reward $s \in (0, 1]$. For samples with incorrect answers or formatting errors, we assign only a fixed formatting reward $r_{\text{fmt}} \in [0, 1)$, without applying any reasoning redundancy constraints. Consequently, the optimization objective not only emphasizes answer correctness but also explicitly encourages the model to discover more concise and more certain reasoning paths in multilingual settings. The final reward for language l is defined as:

$$r^{(l)} = \begin{cases} \max(0, s - \lambda P_{\text{multi}}^{(l)}), & \text{answer is correct,} \\ r_{\text{fmt}}, & \text{otherwise.} \end{cases} \quad (13)$$

The in-language redundancy term suppresses excessive exploration in single-language reasoning, assigning lower rewards to high-entropy reasoning processes. The excess redundancy and group variance terms further drive alignment toward the most concise language within the group, reducing cross-lingual reasoning discrepancies. This design enables the model to progressively transform high-uncertainty reasoning into more stable and efficient reasoning behavior in multilingual environments.

4 Experiments

4.1 Experimental Setup

Training Data. We use the GSM8K (Cobbe et al., 2021) dataset, which contains 7,473 grade-school math problems, as our training data source. We first use Qwen3-32B (Yang et al., 2025a) to translate the dataset into German, French, Japanese, Russian, and Chinese. The translation quality evaluation is provided in Appendix F. We then combine the data from these five languages to construct the final training set. For a fair comparison, all methods use this mixed dataset. We use the five language dataset as default anchors for training; however, LAE remains effective with alternative anchor sets; see Appendix B.2.

Baselines. **GRPO** (Shao et al., 2024) is a critic-free reinforcement learning approach that estimates advantages by normalizing group rewards across multiple responses to the same prompt. **SEER** (Huang et al., 2025c) is a self-enhancing supervised fine-tuning (SFT) compression framework that integrates Best-of-N sampling with adaptive, dataset-specific length filtering to distill concise yet accurate reasoning traces from self-generated outputs. **Step Entropy** (Li et al., 2025b) proposes a two-stage training approach that empowers LLMs to produce compressed Chain-of-Thought (CoT) reasoning during inference by selectively incorporating [SKIP] tokens. **PEAR** (Huang et al., 2025b) is a phase-aware reward design that incorporates entropy signals into reinforcement learning to penalize excessive exploration during the thinking phase while preserving flexibility in the final answer phase, enabling concise reasoning generation with minimal accuracy loss. Original refers to the results obtained by the base model with zero-shot CoT prompting. To ensure baselines reproducibility, we strictly follow the implementation and hyperparameter settings provided in the paper, modifying only the choice of training dataset.

Backbone Models. We conduct experiments using widely adopted Large Reasoning Models (LRMs), including DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025), Qwen3-4B and Qwen3-8B (Yang et al., 2025a). These models have been extensively used in prior studies and span different parameter scales, enabling a fair and comprehensive comparison. We also include experiments on DeepSeek-R1-Distill-Llama-8B (Guo et al., 2025) in Appendix K.3 to further validate the generality of our findings across different model families.

Implementation Details. In all LAE experiments, we set $s = 1$, $r_{\text{fmt}} = 0$, and $\lambda = 1$. Additional experiments analyzing the effect of varying λ are provided in the Appendix J. For RL training, we use open-source framework ver1¹. The training configuration uses a batch size of 128 and a learning rate of 1×10^{-6} . For more training details, please refer to Appendix E. To fully compare the effectiveness and generalization of our method, we use two evaluation benchmarks: **MGSM** (Shi et al., 2022) which has 250 English problems translated through human annotators in 10 languages and **MMATH** (Luo et al., 2025b) which comprises 374 carefully selected math problems from high-quality sources across 10 languages. We also include experiments on **MMMLU** (Hendrycks et al., 2020) dataset in Appendix K.3 to further validate the generality of our findings across different tasks.

We evaluate performance in two dimensions: accuracy (Acc) and the number of generated tokens (Tok). For MGSM dataset, we set a maximum generation length of 16,384 tokens while for the harder MMATH dataset, we provide the model with 32,768 tokens for generation. Following the evaluation protocol of (Guo et al., 2025), we use sampling with a temperature of 0.6 and a top-p value of 0.95. All results are averaged over 5 independent runs.

4.2 Main Results

As shown in Table 2 and Table 3, LAE achieves notable improvements in both accuracy and token compression. Specifically, Empirical results across 11 languages demonstrate that our proposed method significantly reduces token usage by 30% to 43% during inference on MGSM. The efficiency-accuracy trade-off on the more challenging MMATH benchmark is substantially tighter than that observed on MGSM. We attribute this difference to two complementary factors. First, MMATH

¹<https://github.com/volcengine/ver1>

Method	bn		de		en		es		fr		ja		ru		sw		te		th		zh		$\Delta\text{Acc}\uparrow$	Comp. $\uparrow$
	Acc $\uparrow$	Tok $\downarrow$	Acc $\uparrow$	Tok $\downarrow$	Acc $\uparrow$	Tok $\downarrow$	Acc $\uparrow$	Tok $\downarrow$	Acc $\uparrow$	Tok $\downarrow$	Acc $\uparrow$	Tok $\downarrow$	Acc $\uparrow$	Tok $\downarrow$	Acc $\uparrow$	Tok $\downarrow$	Acc $\uparrow$	Tok $\downarrow$	Acc $\uparrow$	Tok $\downarrow$	Acc $\uparrow$	Tok $\downarrow$		
R1-Distill-Qwen-7B																								
Original	56.48	1109	73.20	489	84.80	518	73.92	472	71.52	493	52.88	718	72.32	565	7.44	1032	24.24	2513	57.04	1161	81.28	1046	–	–
GRPO	66.40	1233	84.32	743	94.88	1104	88.72	833	86.00	781	76.32	957	88.64	848	8.80	2550	39.76	2170	75.28	1095	<u>84.80</u>	1107	12.62	-0.48
SEER	54.00	776	67.76	530	89.04	<u>397</u>	76.80	430	73.76	458	57.20	400	76.48	454	6.04	<u>728</u>	26.64	<u>1561</u>	59.92	787	80.48	649	1.18	<u>0.24</u>
Step Entropy	40.88	572	70.00	388	87.12	422	73.04	<u>365</u>	69.52	<u>372</u>	41.92	464	71.20	<u>453</u>	6.24	1507	24.00	1891	52.96	896	70.72	1084	-4.32	0.17
PEAR	<u>63.20</u>	1003	<u>79.84</u>	<u>341</u>	<u>93.68</u>	440	82.72	431	79.28	480	68.72	<u>403</u>	<u>84.56</u>	487	10.80	868	34.80	2114	68.56	<u>776</u>	85.04	<u>383</u>	<u>8.73</u>	0.23
LAE (ours)	60.16	<u>688</u>	78.40	300	91.36	270	<u>82.80</u>	348	79.60	363	<u>71.20</u>	366	82.32	434	<u>9.84</u>	627	<u>36.40</u>	1320	<u>70.32</u>	418	82.96	285	8.20 $\uparrow$	0.43 $\uparrow$
Qwen3-4B																								
Original	82.72	2645	90.40	1321	95.76	2125	91.68	1292	85.20	1421	83.68	1816	91.12	1548	32.48	4370	72.96	3405	87.20	2068	88.08	2175	–	–
GRPO	<u>82.40</u>	1952	90.72	1085	96.48	1529	<u>93.36</u>	1013	87.52	1123	84.40	1332	<u>91.20</u>	1119	<u>32.80</u>	<u>3641</u>	72.80	2722	87.04	1563	89.04	<u>1578</u>	<u>0.59</u>	0.23
SEER	81.52	<u>1780</u>	89.76	1054	94.96	1337	92.72	<u>991</u>	<u>87.84</u>	<u>1029</u>	<u>84.48</u>	<u>1282</u>	89.68	1331	30.64	3443	71.68	2753	86.56	1395	87.04	2236	-0.40	0.23
Step Entropy	71.28	4090	76.00	2197	<u>96.08</u>	2110	85.20	1944	75.04	2195	63.92	3514	72.80	3208	32.96	4345	<u>74.16</u>	3319	76.80	3880	53.92	6855	-11.19	-0.66
PEAR	82.80	1930	<u>70.00</u>	<u>1004</u>	96.08	1273	92.40	999	88.00	1052	83.76	1294	<u>91.20</u>	<u>1112</u>	32.72	3777	71.60	<u>2513</u>	<u>87.28</u>	<u>1389</u>	89.44	1591	0.43	<u>0.27</u>
LAE (ours)	82.08	1590	<u>90.56</u>	935	96.48	<u>1313</u>	<u>93.76</u>	907	87.60	926	85.84	1125	<u>91.76</u>	949	32.64	3208	75.12	2254	88.56	1256	89.84	1262	1.18 $\uparrow$	0.35 $\uparrow$
Qwen3-8B																								
Original	87.52	2381	90.56	1511	96.56	2096	91.76	1523	88.32	1531	85.12	1842	91.84	1616	60.88	3306	80.40	3081	89.60	2179	89.04	2583	–	–
GRPO	88.56	2028	90.56	1245	<u>96.88</u>	1816	93.20	1330	88.64	1358	<u>86.00</u>	1520	92.48	1424	60.64	2873	<u>81.04</u>	2662	89.76	1818	<u>89.44</u>	2119	<u>0.51</u>	0.15
SEER	87.84	1839	<u>91.44</u>	1165	96.40	1474	<u>93.44</u>	<u>1221</u>	89.52	1215	85.36	1491	<u>92.72</u>	1443	60.48	2634	80.24	2461	89.36	<u>1536</u>	86.48	2613	0.15	0.19
Step Entropy	70.32	2756	82.16	1322	95.92	1694	92.08	1152	80.24	<u>1153</u>	73.76	2064	83.28	1661	61.68	2811	79.52	2422	80.80	2286	72.00	3643	-7.26	0.04
PEAR	88.24	<u>1806</u>	90.96	<u>1136</u>	96.64	1688	93.28	1173	88.88	1208	85.84	<u>1377</u>	92.48	<u>1208</u>	60.08	<u>2627</u>	80.56	<u>2415</u>	<u>90.48</u>	1599	89.36	<u>1939</u>	0.47	<u>0.23</u>
LAE (ours)	<u>88.40</u>	1688	91.92	1065	97.28	<u>1511</u>	94.00	<u>1155</u>	89.04	1076	86.72	1362	92.96	1090	62.80	2294	82.08	2381	90.88	1482	90.96	1490	1.40 $\uparrow$	0.30 $\uparrow$

Table 2: Results on MGSM benchmarks across three LRMs. “Comp.” denotes the compression ratio $(l - l')/l$ where l and l' are the token counts of the baseline and the proposed method, respectively. “Tok” denotes the model’s generated token counts. The best results in each language are in **bold**, and the second-best results are underlined. $\uparrow$ indicate that LAE significantly outperforms strongest baseline PEAR at $p < 0.01$ levels (two-sample t-test, $n=5$ independent runs).

Method	ar		en		es		fr		ja		ko		pt		th		vi		zh		$\Delta\text{Acc}\uparrow$	Comp. $\uparrow$
	Acc $\uparrow$	Tok $\downarrow$	Acc $\uparrow$	Tok $\downarrow$	Acc $\uparrow$	Tok $\downarrow$	Acc $\uparrow$	Tok $\downarrow$	Acc $\uparrow$	Tok $\downarrow$	Acc $\uparrow$	Tok $\downarrow$	Acc $\uparrow$	Tok $\downarrow$	Acc $\uparrow$	Tok $\downarrow$	Acc $\uparrow$	Tok $\downarrow$	Acc $\uparrow$	Tok $\downarrow$		
R1-Distill-Qwen-7B																						
Original	76.80	4737	88.70	5134	86.03	4447	86.10	4481	75.00	4953	79.01	5252	84.56	4351	77.67	5523	78.14	4569	82.29	5691	–	–
GRPO	78.34	4394	89.04	<u>4253</u>	88.24	4301	86.10	4253	81.82	4094	83.42	<u>4153</u>	87.17	4168	82.09	4821	85.56	4140	84.76	4951	3.22	0.11
SEER	74.40	3416	87.90	4151	83.56	3630	79.08	3340	64.84	<u>3575</u>	78.28	3966	81.48	3385	75.13	4978	70.79	2987	82.55	5411	-3.63	0.21
Step Entropy	71.26	5227	87.50	5268	85.43	4652	84.02	4421	70.59	6823	73.80	5866	84.49	4476	69.79	7934	75.67	4404	81.28	7893	-3.05	-0.15
PEAR	76.20	4140	<u>90.31</u>	4886	86.03	<u>4140</u>	85.09	4279	77.21	<u>4013</u>	<u>80.21</u>	4531	85.56	4253	78.41	<u>4769</u>	78.81	4115	84.96	<u>4582</u>	0.85	0.11
LAE (ours)	<u>76.34</u>	<u>3568</u>	90.37	5208	<u>86.36</u>	4465	<u>85.83</u>	4196	<u>79.41</u>	3529	78.48	4394	<u>86.10</u>	<u>3997</u>	<u>79.41</u>	4194	79.14	3827	84.49	4120	<u>1.16</u> $\uparrow$	<u>0.15</u> $\uparrow$
Qwen3-4B																						
Original	87.17	5471	92.65	6502	91.71	5449	92.05	5395	90.17	5748	89.30	5718	92.31	5382	88.70	5944	90.78	5386	86.83	7362	–	–
GRPO	<u>86.43</u>	4799	93.45	5667	<u>90.98</u>	5012	<u>91.58</u>	5011	<u>90.44</u>	5121	88.90	5100	92.25	4902	<u>88.70</u>	5133	<u>90.51</u>	4796	<u>89.17</u>	<u>5587</u>	<u>0.05</u>	0.12
SEER	86.03	4275	91.64	4995	90.44	4496	90.71	4281	88.97	<u>4568</u>	87.70	<u>4587</u>	90.37	4270	87.10	4591	89.10	<u>4235</u>	85.76	6816	-1.38	<u>0.20</u>
Step Entropy	86.63	5618	<u>93.05</u>	6592	90.84	5697	62.30	14219	88.30	6229	89.10	6036	91.98	5508	88.57	6119	89.71	5385	80.68	10379	-4.05	-0.23
PEAR	85.90	4450	92.58	5320	90.78	4591	90.91	4386	90.04	4801	88.97	4716	<u>92.05</u>	4454	88.03	4852	89.10	4496	88.03	5849	-0.53	0.18
LAE (ours)	86.63	4183	92.78	<u>5034</u>	91.31	<u>4503</u>	91.71	<u>4351</u>	90.64	<u>4881</u>	<u>89.04</u>	<u>4479</u>	<u>91.58</u>	<u>4385</u>	89.04	<u>4639</u>	90.64	4199	88.43	4800	0.11 $\uparrow$	0.13 $\uparrow$
Qwen3-8B																						
Original	89.17	5855	93.72	6708	92.78	5907	92.78	5766	90.11	6256	91.44	5798	93.11	5825	91.11	6215	91.78	5848	88.77	7448	–	–
GRPO	<u>89.24</u>	5426	<u>93.58</u>	6298	93.45	5441	<u>93.45</u>	5285	<u>90.17</u>	5696	91.38	5449	92.38	5481	91.24	5886	<u>91.84</u>	5482	<u>89.71</u>	6947	<u>0.17</u>	0.07
SEER	88.37	4696	92.51	5447	92.18	4812	92.45	4644	88.90	5064	89.84	4834	92.71	4651	90.04	4963	90.91	4826	86.03	7406	-1.08	0.17
Step Entropy	87.70	5493	93.32	6451	93.05	5513	87.97	5822	87.70	6360	90.64	5789	92.51	5517	89.84	5819	90.64	5791	84.49	8246	-1.69	0.02
PEAR	89.04	5186	93.72	6014	<u>93.25</u>	5274	93.05	5106	89.77	5575	91.64	5250	<u>92.85</u>	5137	<u>91.11</u>	5511	91.38	5164	89.44	<u>6674</u>	0.05	0.11
LAE (ours)	89.48	<u>5075</u>	<u>93.58</u>	<u>5779</u>	92.78	<u>5129</u>	93.58	4918	90.20	<u>5367</u>	<u>91.44</u>	<u>4897</u>	93.58	<u>5029</u>	90.55	<u>5269</u>	92.07	<u>4958</u>	91.35	5906	0.39 $\uparrow$	<u>0.15</u> $\uparrow$

Table 3: Results on MMATH benchmarks across three LRMs. The best results in each language are in **bold**, and the second-best results are underlined. $\uparrow$ indicate that LAE significantly outperforms strongest baseline PEAR at $p < 0.01$ levels (two-sample t-test, $n=5$ independent runs).

problems are sourced from more advanced mathematical domains, where longer reasoning chains reflect genuine logical necessity rather than exploratory redundancy. In such cases, the entropy

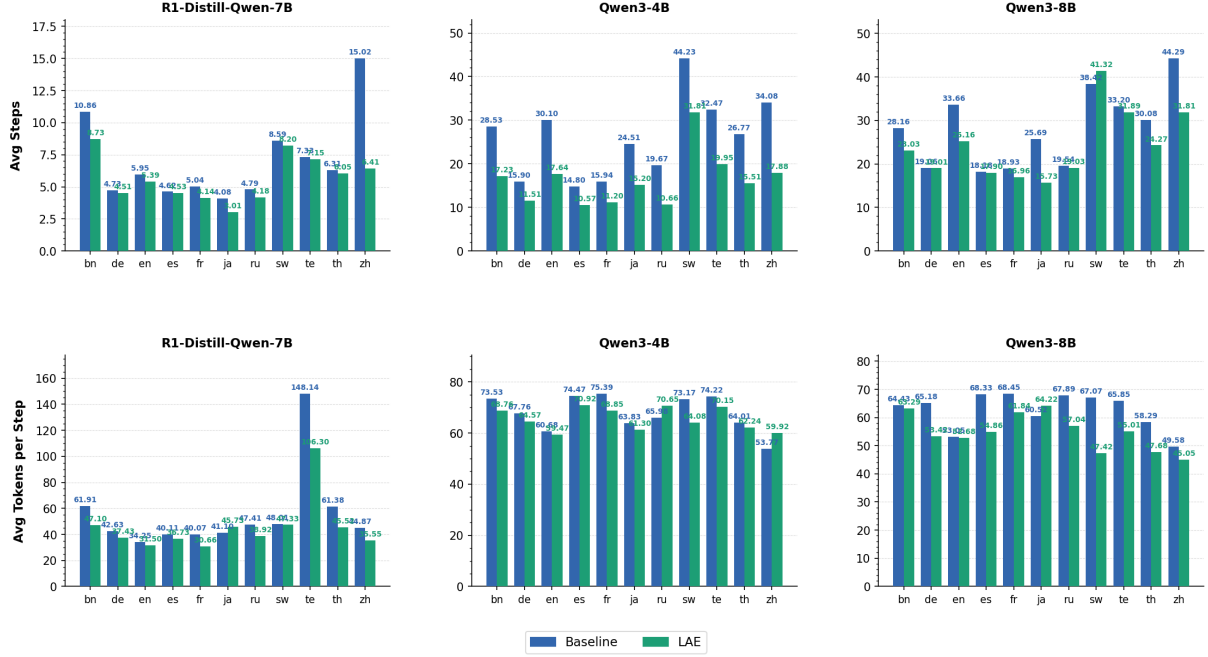


Figure 2: Step changes across three models before and after training on MGSM dataset. For most models and languages, LAE reduces the number of reasoning steps primarily in the think phase, indicating that compression is mainly achieved by eliminating redundant reasoning steps.

4.3 Token Reduction Analysis

To better understand how LAE achieves token compression, we decompose the generated tokens into two complementary factors: the number of reasoning steps and the number of tokens per step. Figure 2 reports both metrics before and after LAE training on MGSM dataset.

Step reduction is the primary source of compression. Across all models, LAE substantially reduces the number of reasoning steps, while changes in tokens per step remain relatively small. For example, on Qwen3-4B, the average step count in Chinese decreases from 34.08 to 17.88, a 47.6% reduction, whereas tokens per step change only slightly from 53.77 to 59.92. Similar trends are observed for Qwen3-8B and R1-Distill-Qwen-7B. This suggests that LAE mainly improves efficiency by removing redundant reasoning steps rather than shortening the wording within each step. Such behavior is consistent with the entropy-gap reward, which discourages unnecessary verification or repeated derivation once a valid solution path has been identified. The effect is particularly pronounced in low-resource and high-verbosity languages. Languages such as Swahili, Telugu, and Bengali exhibit higher baseline step counts, reflecting increased uncertainty during reasoning. For instance, Swahili reaches an average of 44.23 steps

on Qwen3-4B. After applying LAE, this number is reduced to 31.81, while Telugu decreases from 32.47 to 19.95. These results suggest that the entropy-gap reward is especially effective in scenarios with larger cross-lingual uncertainty disparities, where it suppresses excessive exploration. Although step reduction dominates, tokens per step also show modest but consistent decreases. This suggests that the entropy-based objective further encourages more decisive token-level generation within each retained step. The effect is more evident in R1-Distill-Qwen-7B, where, for example, Telugu decreases from 148.14 to 106.30 tokens per step, compared to smaller changes in the Qwen3 models.

The compression effect generalizes to unseen languages during training. Language such as Bengali show reductions in reasoning steps after training. Bengali decreases from 28.16 to 23.03 steps on Qwen3-8B. This indicates that LAE captures a language-agnostic notion of reasoning efficiency, rather than relying on language-specific patterns, which is consistent with the cross-lingual transfer results in the main experiments.

These findings demonstrate that LAE primarily achieves token compression by eliminating redundant reasoning steps, especially those associated with unnecessary verification and re-exploration. It

Language	Base (s)	LAE (s)	Reduction
bn	369	321	13.0%
de	207	244	-17.9%
en	251	67	73.3%
es	254	121	52.4%
fr	245	196	20.0%
ja	326	215	34.0%
ru	277	273	1.4%
sw	485	312	35.7%
te	1142	550	51.8%
th	475	173	<u>63.6%</u>
zh	338	128	62.1%
Overall	4369	2600	40.5%

Table 4: Per-language wall-clock inference latency of LAE and the R1-Distill-Qwen-7B base model on MGSM. Each language contains 250 samples, with five generation runs per sample. The latency reduction is computed as $(T_{\text{base}} - T_{\text{LAE}})/T_{\text{base}}$.

maintains the token density of essential reasoning while reducing verbosity driven by uncertainty.

4.4 Inference Latency Analysis

While token reduction provides an indication of the compactness of generated reasoning, it does not necessarily imply proportional savings in actual inference time. To directly assess the practical efficiency gains of LAE, we therefore measure the wall-clock inference latency using vLLM. We evaluate R1-Distill-Qwen-7B on all 11 languages of MGSM, with 250 samples per language and five generation runs for each sample. As shown in Table 4, LAE reduces the total inference latency from 4,369 s to 2,600 s, corresponding to a 40.5% reduction. Moreover, LAE achieves lower latency in 10 out of 11 languages, with particularly substantial reductions observed for English (73.3%), Thai (63.6%), Chinese (62.1%), Spanish (52.4%), and Telugu (51.8%). These results demonstrate that the shorter reasoning traces produced by LAE translate into substantial end-to-end inference-time savings, providing direct evidence of practical efficiency improvements beyond token-count reduction.

4.5 Comparison to Adaptive Token-Budget Methods

To examine whether our findings extend beyond entropy-based compression, we further consider adaptive token-budget methods, including SelfBudgeter (Li et al., 2025c), AdaR1 (Luo et al., 2025a),

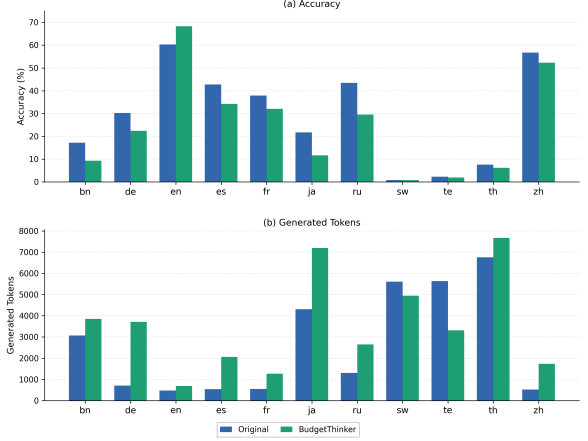


Figure 3: Comparison between the original model and BudgetThinker on multilingual MGSM. BudgetThinker improves English accuracy but generally degrades accuracy and increases generation length in other languages.

and BudgetThinker (Wen et al., 2025). Both SelfBudgeter and AdaR1 derive their budget or length preferences from predominantly English training distributions, making them language-agnostic in architecture but language-dependent in calibration.

We directly evaluate the released BudgetThinker model on multilingual MGSM without additional fine-tuning. As shown in Figure 3, BudgetThinker improves English accuracy but yields an average accuracy drop of 4.73 points across the 11 languages, while increasing token usage by an average of 116% across languages. These results suggest that the limitation is not specific to entropy-based signals, but more generally arises when compression signals are calibrated on a single-language distribution.

5 Conclusion

We present LAE, a multilingual CoT compression framework that extends GRPO with multilingual group sampling and a language-aware entropy-gap reward. By leveraging cross-lingual uncertainty, LAE reduces redundant reasoning while preserving necessary reasoning complexity. Experiments on MGSM and MMATH show 30%–43% token reduction with consistent accuracy improvements across models and languages, including low-resource and unseen languages. These results demonstrate that cross-lingual uncertainty is an effective signal for reducing language-induced redundancy and improving multilingual reasoning efficiency.

Limitations

Although our study focuses on Qwen-architecture models and mathematical reasoning tasks, these design choices are intentional and aligned with the goals of this work. Qwen-based models are known for their strong multilingual capabilities, which makes them a suitable and challenging testbed for investigating multilingual chain-of-thought compression, providing a meaningful validation of our approach. Our entropy-gap analysis suggests that LAE primarily removes redundant reasoning steps, our evaluation remains centered on outcome-level metrics such as accuracy, token reduction, and step reduction, and do not perform fine-grained human evaluation on intermediate reasoning trajectories. Investigating the relationship between entropy-based compression and human judgments of reasoning quality is an important direction for future work.

Ethics Statement

We utilize the open and publicly available benchmarks in our experiments. Additionally, we use the open-resourced LLMs for post-training. Since all the queries are objective math problems, our analysis and results were conducted with sensitivity, and do not involve or affect attributes related to gender, sex, or race, which does not explicitly address ethical considerations in this work.

Acknowledgments

This work is supported by the Fundamental Research Funds for the Central Universities of China under Grant 2024JBGP008 and the National Natural Science Foundation of China (No. 62406018, 62476023, 61976016, 62376019), and the authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper.

References

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, and 1 others. 2024. Do not think that much for $2+3=?$ on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers

to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Changhao Guan, Chao Huang, Hongliang Li, You Li, Ning Cheng, Ziheng Liu, Yufeng Chen, Jinan Xu, and Jian Liu. 2025. Multi-stage llm fine-tuning with a continual learning setting. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5499–5513.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Chao Huang, Yufeng Chen, Changhao Guan, Guang Yang, Dongze Chen, and Kaiyu Huang. 2026. Disentangled contrastive learning for zero-shot multilingual dense retrieval. *arXiv preprint arXiv:2608.02189*.

Chao Huang, Fengran Mo, Yufeng Chen, Changhao Guan, Zhenrui Yue, Xinyu Wang, Jinan Xu, and Kaiyu Huang. 2025a. Boosting data utilization for multilingual dense retrieval. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 12373–12389.

Chen Huang, Wei Lu, and Wenxuan Zhang. 2025b. Pear: Phase entropy aware reward for efficient reasoning. *arXiv preprint arXiv:2510.08026*.

Kerui Huang, Shuhan Liu, Xing Hu, Tongtong Xu, Lingfeng Bao, and Xin Xia. 2025c. Reasoning efficiently through adaptive chain-of-thought compression: A self-optimizing framework. *arXiv preprint arXiv:2509.14093*.

Shulin Huang, Yiran Ding, Junshu Pan, and Yue Zhang. 2025d. Beyond english-centric training: How reinforcement learning improves cross-lingual reasoning in llms. *arXiv preprint arXiv:2509.23657*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Lingjie Jiang, Xun Wu, Shaohan Huang, Qingxiu Dong, Zewen Chi, Li Dong, Xingxing Zhang, Tengchao Lv, Lei Cui, and Furu Wei. 2025. Think only when you need with large hybrid-reasoning models. *arXiv preprint arXiv:2505.14631*.

Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. 2025. C3ot: Generating shorter chain-of-thought without compromising effectiveness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24312–24320.

- 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.
- Abhinav Kumar, Jaechul Roh, Ali Naseh, Marzena Karpinska, Mohit Iyyer, Amir Houmansadr, and Eugene Bagdasarian. 2025. Overthink: Slow-down attacks on reasoning llms. *arXiv preprint arXiv:2502.02542*.
- Chengzhengxu Li, Xiaoming Liu, Zhaohan Zhang, Shaochu Zhang, Shengchao Liu, Guoxin Ma, Yu Lan, and Chao Shen. 2025a. Upfront chain-of-thought: A cooperative framework for chain-of-thought compression. *arXiv preprint arXiv:2510.08647*.
- Zeju Li, Jianyuan Zhong, Ziyang Zheng, Xiangyu Wen, Zhijian Xu, Yingying Cheng, Fan Zhang, and Qiang Xu. 2025b. Compressing chain-of-thought in llms via step entropy. *arXiv preprint arXiv:2508.03346*.
- Zheng Li, Qingxiu Dong, Jingyuan Ma, Di Zhang, Kai Jia, and Zhifang Sui. 2025c. Selfbudgeter: Adaptive token allocation for efficient llm reasoning. *arXiv preprint arXiv:2505.11274*.
- Haotian Luo, Haiying He, Yibo Wang, Jinluan Yang, Rui Liu, Naiqiang Tan, Xiaochun Cao, Dacheng Tao, and Li Shen. 2025a. Adar1: From long-cot to hybrid-cot via bi-level adaptive reasoning optimization. *arXiv e-prints*, pages arXiv–2504.
- Wenyang Luo, Wayne Xin Zhao, Jing Sha, Shijin Wang, and Ji-Rong Wen. 2025b. Mmath: A multilingual benchmark for mathematical reasoning. *arXiv preprint arXiv:2505.19126*.
- Ziqing Qiao, Yongheng Deng, Jiali Zeng, Dong Wang, Lai Wei, Guanbo Wang, Fandong Meng, Jie Zhou, Ju Ren, and Yaoyue Zhang. 2025. Concise: Confidence-guided compression in step-by-step efficient reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 8021–8040.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, and 1 others. 2022. Language models are multilingual chain-of-thought reasoners. *arXiv preprint arXiv:2210.03057*.
- Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.
- 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*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Hao Wen, Xinrui Wu, Yi Sun, Feifei Zhang, Liye Chen, Jie Wang, Yunxin Liu, Yunhao Liu, Ya-Qin Zhang, and Yuanchun Li. 2025. Budgetthinker: Empowering budget-aware llm reasoning with control tokens. *arXiv preprint arXiv:2508.17196*.
- Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. 2025. When more is less: Understanding chain-of-thought length in llms. *arXiv preprint arXiv:2502.07266*.
- Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. 2025. Tokenskip: Controllable chain-of-thought compression in llms. *arXiv preprint arXiv:2502.12067*.
- Jianzhi Yan, Le Liu, Youcheng Pan, Shiwei Chen, Zike Yuan, Yang Xiang, and Buzhou Tang. 2025. From long to lean: Performance-aware and adaptive chain-of-thought compression via multi-round refinement. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 12290–12306.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. 2025b. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*.
- Guang Yang, Changhao Guan, Chao Huang, Yufeng Chen, and Kaiyu Huang. 2026. [Adaptive utilization of low-rank adaptation via conditioned gating](#). In *Forty-third International Conference on Machine Learning*.
- Hang Yuan, Bin Yu, Haotian Li, Shijun Yang, Christina Dan Wang, Zhou Yu, Xueyin Xu, Weizhen Qi, and Kai Chen. 2025. Not all tokens are what you need in thinking. *arXiv preprint arXiv:2505.17827*.

- Wenhao Zeng, Yaoning Wang, Chao Hu, Yuling Shi, Chengcheng Wan, Hongyu Zhang, and Xiaodong Gu. 2025. Pruning the unsurprising: Efficient code reasoning via first-token surprisal. *arXiv preprint arXiv:2508.05988*.
- Xue Zhang, Yunlong Liang, Fandong Meng, Songming Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. 2025a. [CM-align: Consistency-based multilingual alignment for large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 25689–25702, Suzhou, China. Association for Computational Linguistics.
- Xue Zhang, Yunlong Liang, Fandong Meng, Songming Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. 2025b. [Less, but better: Efficient multilingual expansion for LLMs via layer-wise mixture-of-experts](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17948–17963, Vienna, Austria. Association for Computational Linguistics.
- Xue Zhang, Yunlong Liang, Fandong Meng, Songming Zhang, Kaiyu Huang, Yufeng Chen, Xu Jinan, and Jie Zhou. 2026. [Think natively: Unlocking multilingual reasoning with consistency-enhanced reinforcement learning](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 20766–20783, San Diego, California, United States. Association for Computational Linguistics.
- Shangzhiqi Zhao, Jiahao Yuan, Guisong Yang, and Usman Naseem. 2025. Can pruning improve reasoning? revisiting long-cot compression with capability in mind for better reasoning. *arXiv preprint arXiv:2505.14582*.
- Zihao Zhu, Hongbao Zhang, Ruotong Wang, Ke Xu, Siwei Lyu, and Baoyuan Wu. 2025. To think or not to think: Exploring the unthinking vulnerability in large reasoning models. *arXiv preprint arXiv:2502.12202*.

A More Analysis of Results

Results on MGSM. As summarized in Table 2, LAE demonstrates consistent and substantial improvements in both accuracy and token efficiency. Empirical results on 11 languages show that the proposed method reduces token usage during inference by 30%–43%, while simultaneously achieving performance gains. This indicates that our design effectively eliminates redundant computations, optimizing inference efficiency. These findings suggest that our sampling and reward mechanisms effectively incentivize the model to internalize the most parsimonious reasoning paths across the five primary languages. Furthermore, this learned capability generalizes robustly to languages not encountered during the training phase, highlighting the cross-lingual transferability of our approach.

In comparison to GRPO, LAE offers advantages in handling language-specific reasoning redundancy. While GRPO leverages entropy-based rewards to improve accuracy, its performance often becomes inconsistent when scaling across multiple languages, particularly with respect to compression efficiency. In contrast, LAE utilizes a multilingual relative entropy gap that more effectively adjusts the reasoning process across languages. For instance, on the MGSM benchmark, our proposed method achieves competitive performance compared to PEAR, yielding an accuracy of 8.20% versus PEAR’s 8.73%. Crucially, our approach demonstrates significantly higher compression efficiency, attaining a compression ratio of 43%, which is nearly double that of PEAR (23%), while still maintaining a robust accuracy of 8.20%. This result indicates a high level of token efficiency, as substantial accuracy improvements are achieved without generating an excessive number of tokens. Importantly, these accuracy gains are obtained without sacrificing token reduction, highlighting LAE’s ability to effectively balance accuracy and compression objectives.

Furthermore, LAE adapts well to languages with varying resource levels, addressing the challenges of low-resource languages. For example, languages such as Bengali (bn), Swahili (sw), and Telugu (te), which typically require more tokens due to their linguistic structures, benefit from the relative entropy gap mechanisms employed by LAE, resulting in improved efficiency while preserving reasoning accuracy. This is particularly important in managing the accuracy of reasoning while reducing

unnecessary token expansion.

In terms of multilingual input-output behavior, we observe that high-resource languages such as English, Spanish, French tend to have shorter reasoning paths, whereas low-resource languages like Bengali, Swahili, Telugu produce longer chains of thought due to differences in syntax, morphology, and resource availability. LAE effectively addresses these differences by suppressing redundant reasoning steps, thereby improving both the compression ratio and overall performance without sacrificing reasoning accuracy.

Results on MMATH. As shown in Table 3, the efficiency–accuracy trade-off on the more challenging MMATH benchmark is substantially tighter than that observed on MGSM. Nevertheless, language-aware, group-level redundancy control yields more token-efficient performance gains than correctness-only reinforcement learning or monolingual compression approaches.

Across all three LRMs, LAE consistently improves average accuracy while reducing generation length. For R1-Distill-Qwen-7B, LAE achieves a 1.16 accuracy improvement with 15% compression, outperforming PEAR (0.85 accuracy gain, 11% compression) and avoiding the performance degradation observed with SEER and Step-Entropy, both of which result in negative accuracy changes. For Qwen3-4B, several baseline methods exhibit unstable behavior, trading compression for degraded generation quality. In particular, Step-Entropy induces repetitive generation in fr, leading to substantial token inflation. In contrast, LAE remains stable and attains the highest compression rate among non-degrading methods (23%), while still yielding a modest but positive accuracy gain of 0.11. For Qwen3-8B, LAE again demonstrates a more favorable accuracy–compression trade-off, delivering a 0.39 accuracy improvement with 15% compression. This result exceeds those of GRPO (0.17 accuracy gain, 7% compression) and PEAR (0.05 accuracy gain, 11% compression).

A language-level analysis further reveals that the gains from LAE do not arise from uniform shortening across all outputs. Instead, LAE selectively suppresses over-exploratory reasoning in languages where the model exhibits higher baseline verbosity, while maintaining or improving correctness. This effect is particularly pronounced in high-verbosity languages, where substantial token reductions are accompanied by accuracy recovery

rather than degradation.

Overall. Experiments on MGSM and MMATH consistently show that LAE’s combination of multilingual group sampling and relative entropy-gap reward provides an effective mechanism for language-aware redundancy suppression. On MGSM, LAE yields substantial token savings, achieving approximately 30% to 43% compression depending on the model and language, while simultaneously improving average accuracy and generalizing to unseen languages in training. These results suggest that LAE learns transferable and more parsimonious reasoning patterns, rather than merely shortening model outputs. On the more challenging MMATH benchmark, where the trade-off between efficiency and accuracy is tighter, LAE remains stable across three LRMs and achieves the best or near-best compression among methods that do not degrade accuracy, while still delivering positive accuracy gains. In contrast, correctness-only reinforcement learning and monolingual compression baselines frequently exhibit instability or accuracy drops. Importantly, the observed improvements do not arise from uniform truncation. Instead, LAE selectively suppresses over-explorative, high-entropy reasoning that is more prevalent in verbose and often lower-resource languages, while preserving necessary uncertainty in the answer phase. This selective behavior aligns cross-lingual reasoning complexity and improves token efficiency without sacrificing correctness.

B Anchor-Language Selection and Sensitivity Analysis

B.1 Selection Rationale

The selection of languages for multilingual group sampling is designed to balance signal reliability and linguistic diversity. We adopt a strategy of using “Anchor Languages” for training. The core rationale is that the group-level statistics (specifically E_{min} in Equation 6) require a high probability of containing at least one correct and concise reasoning trajectory within each group to serve as a valid reference (anchor) for alignment. Including languages with extremely low baseline accuracy (<10%) would result in a high proportion of “all-zero-reward” groups, which provide no learning signal for the relative entropy gap calculation. Therefore, we curate a set of linguistically diverse yet high-resource languages (German,

Anchor Set	$\Delta\text{Acc} \uparrow$	Comp. $\uparrow$
de, fr, ja, ru, zh (default)	8.20	43%
en, fr, ja, ru, zh	5.20	51%
de, th, ja, ru, zh	6.35	47%

Table 5: Sensitivity of LAE to the choice of anchor languages on MGSM using R1-Distill-Qwen-7B. Each perturbed setting replaces one language in the default anchor set.

French, Japanese, Russian, and Chinese) as anchors. These languages cover distinct syntactic and morphological families, providing a robust reference frame for reasoning compression. Moreover, [Huang et al. \(2025d\)](#) show that the generalization performance of the GRPO algorithm varies across languages, with several non-English languages outperforming English. Therefore, English is excluded from the default anchor set. Crucially, the goal of LAE is not merely multilingual co-training, but cross-lingual reasoning transfer. As demonstrated in Table 2, this approach allows the model to learn parsimonious reasoning patterns that generalize effectively to unseen and low-resource languages (e.g., Swahili, Bengali), even though they are not present in the training mix. This suggests that the model learns a language-agnostic reasoning efficiency from the anchor languages.

B.2 Sensitivity to Anchor-Language Selection

To examine whether LAE critically depends on the particular anchor-language set, we performed a sensitivity analysis on R1-Distill-Qwen-7B using MGSM. Starting from our default anchor set {de, fr, ja, ru, zh}, we consider two perturbations: replacing German with English, and replacing French with Thai. These settings respectively test the effect of including English and using a lower-resource language as an anchor.

The results show that LAE remains effective under both anchor perturbations. Replacing German with English increases the compression ratio from 43% to 51%, while still achieving a substantial accuracy gain of 5.20 points. Replacing French with Thai likewise preserves a strong accuracy gain of 6.35 points and achieves 47% compression. These results demonstrate that the effectiveness of LAE does not depend on a single fixed anchor language configuration.

Nevertheless, the default anchor set provides the best accuracy and compression trade off among the three configurations. Using English as an anchor

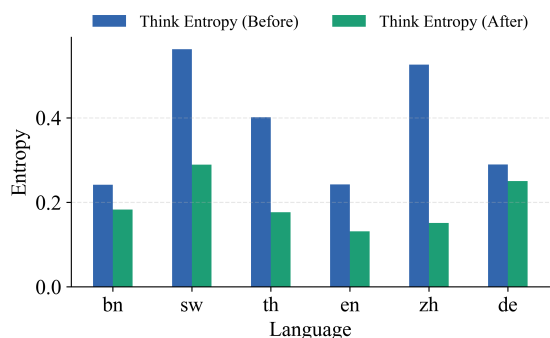


Figure 4: Entropy change of R1-Distill-Qwen-7B model before and after training. The first three languages (bn, sw, th) are considered low-resource languages, while the last three languages (en, zh, de) are medium to high-resource languages.

leads to stronger compression but a smaller accuracy improvement. This observation is consistent with our motivation for excluding English from the default anchor set. Its relatively low baseline entropy can make it disproportionately influential in determining $E_{\min}$, thereby encouraging more aggressive cross lingual compression. Replacing French with Thai results in a smaller but still noticeable reduction in the accuracy gain, which is consistent with the weaker reference signal provided by a lower resource language.

Overall, these results suggest that LAE is robust to reasonable changes in the anchor language set, while the proposed selection of linguistically diverse and relatively high resource languages without English provides the most favorable accuracy and compression trade off in our experiments. This supports our original design rationale and further demonstrates that LAE does not rely on a narrowly tuned anchor language configuration.

C Entropy Change of LAE on Reasoning

As illustrated in Figure 4, we investigate entropy reduction during the thinking phase. After training, thinking entropy decreased across all languages, particularly those with higher initial uncertainty (e.g., Bengali (bn), Thai (th), and Swahili (sw)), indicating a shift in the model’s reasoning. Notably, this reduction extends to both seen and unseen languages, demonstrating LAE’s generalizability and suggesting that its entropy-based reward optimizes reasoning across both trained and zero-shot languages.

D Redundant Reasoning Associated with a Large Entropy Gap

Question: 伊莉莎每周工作的前 40 个小时的时薪是 10 美元。她加班还能获得标准时薪 1.2 倍的工资。如果 Eliza 这周工作了 45 小时，她这周的收入是多少？

Ground Truth

460

Think

好的，我现在来仔细思考一下这个问题，希望我能一步一步地解决它。问题描述是这样的：伊莉莎每周工作的前40个小时的时薪是10美元。她加班的话，还能获得标准时薪1.2倍的工资。这周她工作了45小时，那她的收入是多少呢？... 现在，把正常工作时间和加班时间的工资加起来，就是她这周的总收入。400美元加上60美元，总共是460美元。所以，伊莉莎这周的收入应该是460美元。

不过，我觉得有必要再检查一下计算是否正确，避免出现低级错误。... 加起来的话，400+60=460美元。没错，计算是对的。

另外，我再想一下，是否有可能题目中的“标准时薪”指的是包括加班在内的时薪，但通常情况下，加班工资是单独计算的，所以这里的理解应该是正确的。... 嗯，看来我的思路是对的，没有问题。那她的总收入就是460美元。

答案

伊莉莎这周的收入是460美元。

Answer

伊莉莎这周工作了45小时，其中前40小时是正常工作时间，每小时10美元，加班5小时，每小时的工资是标准时薪的1.2倍，即12美元每小时。

1. 正常工作时间工资计算：40小时 × 10美元/小时 = 400美元

2. 加班工资计算：5小时 × 12美元/小时 = 60美元

3. 总收入计算：400美元 + 60美元 = 460美元

最终，伊莉莎这周的收入是：460

Thinking token: 605

Answer token: 207

Thinking entropy: 0.692

Answer entropy: 0.100

Entropy gap: 0.592

In this example from the MGSM dataset, the R1-Distill-Qwen-7B model exhibits redundant reasoning behavior during the solution process. After initially deriving the correct answer (460), the model continues to perform two additional verification steps. We observe that these verification steps are associated with relatively high thinking entropy, indicating increased uncertainty in the model’s intermediate reasoning states. This leads to unnecessary exploration despite the correct answer already being obtained. For changes in entropy during the thinking phase before and after training, please refer to Appendix C.

E Implementation Details

All experiments are conducted on multiple NVIDIA A100-SXM4-80GB GPUs. Table 6 summarizes the detailed experimental settings.

Settings	Value
<i>Training Setup</i>	
Total epochs	3
Train batch size	128
Max prompt length	1,536
Max response length	4,096
Learning rate	1×10^{-6}
<i>Rollout</i>	
Rollout engine	vLLM
Temperature	0.6
Top- p	0.95

Table 6: Training Settings

F Translation Quality Evaluation

To verify that the performance gains reported in our experiments are not artifacts of poor translation quality, we conducted a systematic human evaluation of the translated training data. Specifically, we collaborated with a professional data annotation service to recruit linguistic experts proficient in each of the five target languages (German, French, Japanese, Russian, and Chinese). Each annotator was asked to evaluate 50 randomly sampled problem translations per language on a five-point scale defined as follows:

- **5 – Excellent:** The translation is fully accurate, mathematically precise, and natural. All meanings, constraints, and nuances are correctly preserved with no errors.

Lang	Score 3	Score 4	Score 5
German	1/50 (2.0%)	7/50 (14.0%)	42/50 (84.0%)
French	2/50 (4.0%)	2/50 (4.0%)	46/50 (92.0%)
Japanese	0/50 (0.0%)	2/50 (4.0%)	48/50 (96.0%)
Russian	4/50 (8.0%)	4/50 (8.0%)	42/50 (84.0%)
Chinese	1/50 (2.0%)	2/50 (4.0%)	47/50 (94.0%)

Table 7: Human evaluation results of translation quality across five training languages. Each language is evaluated on 50 randomly sampled problems. Scores range from 1 (Incorrect) to 5 (Excellent).

- **4 – Good:** The translation is mostly accurate, with only minor issues (e.g., slight awkward phrasing or minor terminology inconsistency) that do not affect understanding.
- **3 – Fair:** The translation conveys the main idea but contains noticeable issues such as ambiguous wording, partial loss of meaning, or imprecise mathematical expressions.
- **2 – Poor:** The translation has significant errors, including incorrect mathematical terms, missing or altered conditions, or misleading phrasing that affects understanding.
- **1 – Incorrect:** The translation is fundamentally wrong, with major mistranslations, incorrect logic, or critical loss of meaning.

The evaluation results are summarized in Table 7. Across all five languages, 84% to 96% of translations received the highest rating of Excellent (Score 5), and over 92% of all samples received a rating of Good or above (Score 4 or 5). No translations were assigned a rating of Poor or Incorrect. These results confirm that the Qwen3-32B-generated translations are generally accurate, mathematically faithful, and linguistically natural.

Beyond translation accuracy, our training methodology further mitigates the risk that any residual translation noise could influence the observed compression gains. In our multilingual group sampling strategy, each problem is jointly optimized across all five languages simultaneously, and the entropy-gap reward is computed relative to other languages within the same group. This design encourages the model to align reasoning behaviors across semantically equivalent inputs rather than exploit language-specific inconsistencies. Consequently, even if isolated translations contain minor imperfections, the cross-lingual alignment objective prevents the model from benefiting by learning

Method	$\Delta\text{Acc} \uparrow$	Comp. $\uparrow$
LAE	8.20	0.43
w/o. Entropy Gap (P_{Lang})	7.05	0.29
w/o. Relative (P_{Relative})	7.35	0.20
w/o. Var (P_{var})	6.12	0.23

Table 8: Results of Ablation Studies by removing each component of LAE.

to simplify flawed inputs. The human evaluation results and the structure of our training objective jointly support the conclusion that the reported token compression and accuracy improvements reflect genuine gains in multilingual reasoning efficiency.

G Ablation Studies

G.1 Reward Ablation

As shown in Table 8, we analyze the contribution of each component in the proposed multilingual entropy gap reward. The results demonstrate that all three terms are essential and mutually complementary, with each serving a distinct and non-substitutable function.

Removing P_{Lang} leads to a clear degradation in both compression efficiency, dropping from 43% to 29%, and accuracy gain, decreasing from 8.20 to 7.05. This indicates that multilingual group sampling alone is insufficient without an explicit constraint on redundant uncertainty. In the absence of this penalty, the model tends to revert to longer and more exploratory reasoning patterns within individual languages.

Removing P_{Relative} causes the most significant reduction in compression ratio, from 43% to 20%, along with a noticeable accuracy drop of 0.85 points. Without alignment to the minimum-entropy language within the group, each language independently retains verbose reasoning trajectories. This confirms that P_{Relative} is the primary driver of compression efficiency by leveraging intra-group contrast.

Removing P_{var} results in the largest accuracy decline, from 8.20 to 6.12, while only moderately affecting the compression ratio. This suggests that the variance penalty contributes beyond output shortening. By discouraging cross-lingual inconsistency in reasoning complexity, it prevents imbalanced behaviors where high-resource languages are over-compressed and low-resource languages remain redundant. Such imbalance introduces con-

Method	$\Delta\text{Acc} \uparrow$	Comp. $\uparrow$
GRPO	12.62	-0.48
Multi + 0/1	5.97	-0.14
LAE	8.20	0.43

Table 9: Ablation results on MGSM using R1-Distill-Qwen-7B. “GRPO” denotes monolingual group sampling with binary reward, “Multi + 0/1” denotes multilingual group sampling with binary reward, and “LAE” denotes multilingual group sampling with the proposed entropy-gap reward.

flicting optimization signals and undermines training stability.

Overall, the three components form a hierarchical mechanism. P_{Lang} suppresses excessive exploration within each language, P_{Relative} aligns reasoning toward the most concise trajectory in the group, and P_{var} enforces global consistency across languages. Their interaction is inherently synergistic, as removing any single component disproportionately weakens the effectiveness of the others. This demonstrates that the proposed reward operates as an integrated framework rather than a set of independent heuristics.

G.2 Component Ablation

To further clarify the motivation for multilingual group sampling and its relationship to the entropy-gap reward, we present an additional ablation study that isolates the effect of each component. Specifically, we introduce a variant that combines multilingual group sampling with a standard binary correctness reward (0/1 signal), replacing the full LAE reward while keeping all other settings identical. The results on MGSM with R1-Distill-Qwen-7B are summarized in Table 9.

This experiment yields two key findings. First, multilingual group sampling paired with a correctness-only reward fails to achieve compression, instead resulting in token expansion. The underlying reason is that a binary correctness signal treats all valid trajectories equally regardless of their length or entropy profile. Within a multilingual group, a verbose English trajectory and a concise German trajectory receive identical rewards if both produce correct answers, providing no compression incentive. In the absence of an entropy-based signal, the model can instead increase exploratory reasoning in languages of higher-uncertainty to satisfy the correctness objec-

tive, leading to token inflation of approximately 14%.

Second, the full LAE framework outperforms the sampling-only variant in both accuracy (+8.20 vs. +5.97) and compression ratio (+0.43 vs. -0.14). This gap demonstrates that multilingual group sampling and the entropy-gap reward are mutually reinforcing rather than independently effective. Multilingual group sampling is designed to expose cross-lingual reasoning disparities within each optimization step, constructing an intra-group reference structure that makes relative redundancy meaningful. The language-aware entropy-gap reward, in turn, exploits this structure by penalizing each language’s trajectory with respect to the minimum-entropy trajectory within the same group, a signal that only exists because the group contains responses across multiple languages. This relative redundancy penalty (P_{Relative}) is precisely what converts cross-lingual reasoning differences into a compression gradient, explaining why LAE achieves 43% compression while the multilingual group with a 0/1 reward baseline achieves none.

These results confirm that multilingual group sampling and the entropy-gap reward constitute a co-adapted mechanism whose effectiveness is validated both by the main results and by the degradation observed when either component is replaced. The motivation for multilingual grouping therefore lies not in its independent effect, but in the intra-group cross-lingual structure it creates for the entropy-gap reward to exploit.

H Computational Overhead Analysis of Multilingual Group Sampling

A natural concern regarding our multilingual group sampling strategy is whether the training overhead it introduces is proportional to the number of languages used. We address this concern by providing a direct comparison with standard GRPO in terms of total rollout count, which is the primary driver of training computation.

Standard GRPO samples R rollouts per prompt for a dataset of size D , resulting in $D \times R$ total rollouts during training. Our multilingual group sampling, by contrast, generates one rollout per language for each problem, yielding $D \times N \times 1$ total rollouts, where N is the number of languages. When $N = R$, the two methods produce an identical number of rollouts. In our experiments, we use $N = 5$ languages, whereas a typical GRPO

configuration uses $R = 8$ or more rollouts per prompt. Consequently, our method incurs *fewer* total rollouts than standard GRPO when trained on the same dataset, making the computational cost during training comparable to or even lower than the baselines.

Regarding prompt translation, this step constitutes a one-time, offline preprocessing procedure performed prior to training. It does not affect per-step training computation and therefore does not contribute to the runtime training cost. Given these considerations, the inference-time savings of 30% to 43% in token reduction are achieved without a proportional increase in training expenditure, representing a favorable overall cost-benefit tradeoff.

I Language Consistency

To further examine whether LAE introduces cross-lingual interference or output language inconsistency, we conduct an additional language adherence evaluation on the MGSM dataset, measuring whether the generated response language matches the input language across all 11 languages and 3 backbone models. The results demonstrate that LAE does not cause language drift. On the contrary, it consistently maintains or improves language adherence, with particularly notable gains observed in low-resource languages. Specifically, for R1-Distill-Qwen-7B, Bengali adherence improves from 78.80% to 98.80% and Swahili from 74.80% to 91.20%. For Qwen3-4B, Swahili improves from 55.60% to 79.20% and Telugu from 87.60% to 97.60%. For Qwen3-8B, Swahili improves from 74.40% to 82.80%. Meanwhile, high-resource languages such as English, Chinese, German, and French maintain near-perfect adherence both before and after training, with no observable degradation.

This finding is well aligned with the design principles of LAE. Language drift typically arises when a model is uncertain, causing it to fall back on high-resource languages as a form of uncertainty-driven exploration. By penalizing high entropy in the thinking phase through our entropy-gap reward, LAE discourages such exploratory behavior and encourages more decisive, language-consistent token generation. In addition, our multilingual group sampling mechanism provides explicit cross-lingual contrastive signals within each training step, which further reinforces the model’s awareness of the target language identity.

Model	bn	de	en	es	fr	ja	ru	sw	te	th	zh
LAE Accuracy											
R1-Distill-Qwen-7B	98.8	98.8	100.0	98.4	98.0	99.2	93.6	91.2	99.2	93.2	100.0
Qwen3-4B	99.6	100.0	100.0	98.4	99.2	99.6	100.0	79.2	97.6	100.0	100.0
Qwen3-8B	100.0	99.2	100.0	98.0	98.4	99.6	100.0	82.8	97.6	100.0	100.0
Base Model Accuracy											
R1-Distill-Qwen-7B	78.8	100.0	99.6	100.0	99.6	94.8	100.0	74.8	96.0	94.0	100.0
Qwen3-4B	98.4	98.8	100.0	97.6	100.0	99.6	100.0	55.6	87.6	100.0	100.0
Qwen3-8B	100.0	98.0	100.0	98.0	99.6	100.0	100.0	74.4	97.2	100.0	100.0

Table 10: Language consistency accuracy (%) across different models and languages.

J Reward Penalty Sensitivity Analysis

To assess the robustness of our method with respect to the entropy penalty scaling factor λ , which uniformly scales the full multilingual redundancy penalty term P_{multi} , we conduct a sensitivity analysis using R1-Distill-Qwen-7B on the MGSM benchmark. The results are summarized in Table 11.

Several observations emerge from these results. When $\lambda = 0$, the training objective reduces to a pure correctness reward without any entropy penalty. Under this setting, the model achieves a modest accuracy gain of +5.98 but fails to reduce token usage, yielding a negative compression ratio of -0.14 . This confirms that correctness-only supervision is insufficient to suppress redundant reasoning behavior.

As λ increases from 0 to 1.0, both accuracy and compression improve consistently. This trend indicates that the entropy-gap reward provides a complementary training signal beyond correctness supervision, explicitly guiding the model toward more concise reasoning trajectories. At $\lambda = 1.0$, the method achieves the most favorable balance between accuracy and compression, with $\Delta\text{Acc} = 8.20$ and $\text{Comp.} = 0.43$.

When λ is further increased to 1.5, the compression ratio improves only marginally while accuracy begins to decline noticeably. This suggests that an excessively strong entropy penalty may suppress uncertainty that is genuinely necessary for solving more challenging problems, thereby undermining reasoning quality. These results demonstrate that $\lambda = 1.0$ represents a stable and well-motivated default choice, and that the overall method is not overly sensitive to moderate variations in this hyperparameter.

λ	$\Delta\text{Acc} \uparrow$	Comp. $\uparrow$
0	5.98	-0.14
0.5	6.84	0.21
1.0	8.20	0.43
1.5	5.38	0.45

Table 11: Sensitivity analysis of the entropy penalty scaling factor λ on R1-Distill-Qwen-7B (MGSM).

Languages	LAE (Ours)		PEAR	
	$\Delta\text{Acc}\uparrow$	Comp. $\uparrow$	$\Delta\text{Acc}\uparrow$	Comp. $\uparrow$
de	5.46	0.28	5.00	0.13
de, ru, zh	6.05	0.35	5.54	0.18
de, ru, zh, fr, ja	8.20	0.43	8.73	0.23

Table 12: Performance of different training languages.

K Additional Experiments

K.1 Impact of Language Diversity

As shown in Table 12, we investigate the impact of language diversity in the training corpus that is expanded the set of languages used during training, ranging from one to five languages. The results demonstrate that our method consistently benefits from the additional languages, with both accuracy and reasoning compression ratio, achieving nearly twice the compression ratio than PEAR. Moreover, the introduction of additional languages does not introduce noise, instead it provides complementary supervisory signals that the model can effectively exploit. It highlights that our group-level, relative entropy gap regularization is particularly well suited to multilingual settings, which is more effectively exploits cross-lingual reasoning path difference while PEAR primarily benefits from additional data exposure.

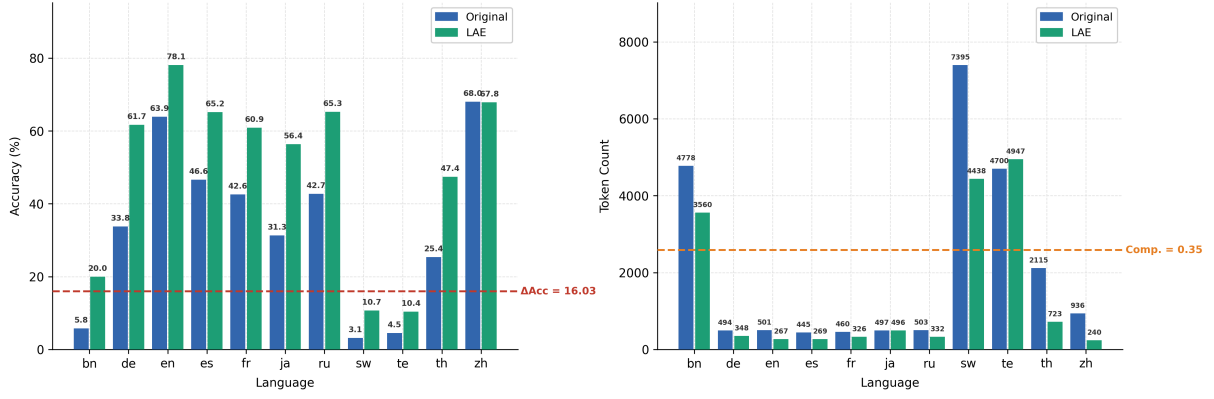


Figure 5: R1-Distill-Llama-8B result on MGSM benchmark. “Comp.” denotes the compression ratio $(l - l')/l$ where l and l' are the token counts of the baseline and the proposed method, respectively.

Dataset Size	LAE (Ours)		PEAR	
	$\Delta\text{Acc}\uparrow$	Comp. $\uparrow$	$\Delta\text{Acc}\uparrow$	Comp. $\uparrow$
1000	2.20	0.15	1.23	0.10
3000	5.40	0.35	2.18	0.16
5000	6.17	0.40	3.02	0.28
7473	8.20	0.43	8.73	0.23

Table 13: Performance comparison on different training data size.

K.2 Effect of Training Data Scale

As shown in Table 13, we further investigate the impact of training data scale on performance. The results show that our method scales more efficiently with data, delivering competitive accuracy while significantly reducing token overhead, especially in resource constraints. Our method exhibits a clear improvement in accuracy as the data budget increases, consistently outperforming PEAR in low- and medium-resource. With only 1k samples, our approach already achieves a 2.20% accuracy gain, and the advantage widens at 3k and 5k samples, where our method reaches 5.40% and 6.17%, compared to PEAR’s 2.18% and 3.02%. More importantly, as data size grows, our method consistently achieves better compression while PEAR shows weaker and less stable compression ratio that drops to 23% with full data, indicating superior scalability and efficiency.

K.3 Additional Dataset and Model

To further assess the generalizability of our method, we conduct additional experiments along two complementary dimensions: architectural diversity and task diversity.

Non-Qwen backbone. We evaluate LAE on R1-Distill-Llama-8B, a model built on the LLaMA

architecture and thus structurally distinct from all Qwen-family models considered in the main experiments. As shown in figure 5, LAE achieves an average accuracy gain of 16.03 points alongside 35% token compression across 11 languages. This result is consistent with our findings on Qwen-family models, demonstrating that the entropy-gap reward mechanism is not tied to any particular model architecture.

Non-mathematical benchmark. We further evaluate all four models on MMMLU, a benchmark spanning 57 diverse subject categories including law, history, science, and computer science. These domains are fundamentally different from the mathematical reasoning tasks on which our models were trained, making MMMLU a suitable testbed for out-of-domain generalization. As shown in Table 14 and Table 15, LAE consistently improves accuracy across all models: 13.05 for R1-Distill-Qwen-7B, 1.69 for Qwen3-8B, 0.35 for Qwen3-4B, and 15.51 for R1-Distill-Llama-8B, while achieving meaningful token compression in all settings. These results confirm that the benefits of LAE extend beyond mathematical reasoning to broader knowledge-intensive tasks.

These supplementary experiments demonstrate that LAE’s effectiveness is neither model-family-specific nor domain-specific. The cross-lingual entropy-gap reward provides a general mechanism for suppressing redundant reasoning chains, one that transfers robustly across different model architectures and task types.

Table 14: Accuracy results on MMMLU benchmark across four LRMs.

Model	ar	bn	de	es	fr	hi	id	it	ja	ko	pt	sw	yo	zh	$\Delta\text{Acc}\uparrow$
R1-Distill-Qwen-7B															
Original	32.07	36.56	26.88	39.09	38.18	25.68	34.39	40.91	36.21	29.82	40.84	7.09	3.72	35.72	—
LAE	41.96	43.79	51.23	54.67	54.04	42.32	46.53	49.12	52.35	34.95	52.00	17.47	11.93	57.47	13.05
Qwen3-4B															
Original	71.44	68.77	74.18	78.11	77.05	68.84	75.09	76.28	76.21	73.96	75.72	25.75	27.58	75.37	—
LAE	70.32	69.33	77.75	78.04	76.07	69.82	76.56	77.40	76.00	73.61	76.28	25.68	26.11	76.28	0.35
Qwen3-8B															
Original	76.63	73.47	77.54	78.39	73.68	71.72	76.63	78.81	78.88	76.70	79.58	39.51	21.75	79.37	—
LAE	77.47	74.74	80.07	80.00	77.12	74.25	78.81	80.84	79.23	76.14	80.35	45.89	21.82	79.65	1.69
R1-Distill-Llama-8B															
Original	22.67	7.09	32.84	39.30	34.67	20.98	27.37	41.89	33.75	37.33	38.88	2.46	3.51	41.75	—
LAE	45.89	25.19	48.49	57.12	49.40	47.79	48.42	58.11	48.07	47.93	54.74	5.33	7.79	57.33	15.51

Table 15: Token count results on MMMLU benchmark across four LRMs.

Model	ar	bn	de	es	fr	hi	id	it	ja	ko	pt	sw	yo	zh	Comp. $\uparrow$
R1-Distill-Qwen-7B															
Original	1280	1116	1071	868	736	1148	903	760	2274	1881	778	1395	1797	1831	—
LAE	881	1344	739	791	624	1007	744	701	948	1967	626	1251	1630	1580	0.15
Qwen3-4B															
Original	1951	2981	1958	1632	1685	2709	1976	1803	2091	2201	1673	7140	3677	3134	—
LAE	1381	2155	1417	1143	1229	1840	1375	1288	1468	1513	1213	6433	3098	1853	0.28
Qwen3-8B															
Original	1939	2975	1970	1752	1810	2923	2047	1810	2150	2113	1806	5524	5117	2881	—
LAE	1584	2355	1580	1476	1533	2226	1646	1517	1632	1704	1508	3576	5327	2148	0.19
R1-Distill-Llama-8B															
Original	1631	2000	816	1076	1011	1841	1148	1070	1471	1326	1092	9244	2793	2670	—
LAE	1290	1828	873	1069	905	1538	1172	1007	1844	1261	930	7398	2565	1906	0.08