Verbosity-Aware Rationale Reduction: Sentence-Level Rationale Reduction for Efficient and Effective Reasoning

Joonwon Jang[1,5] **Jaehee Kim**[2] **Wonbin Kweon**[3] **Seonghyeon Lee**[4] **Hwanjo Yu**[1,*]

POSTECH[1], Seoul National University[2], University of Illinois Urbana-Champaign[3]
Kyungpook National University[4], LG AI Research[5]
{kaoara, hwanjoyu}@postech.ac.kr jaehee_kim@snu.ac.kr
wonbin@illinois.edu sh0416@knu.ac.kr

Abstract

Large Language Models (LLMs) rely on generating extensive intermediate reasoning units (e.g., tokens, sentences) to enhance final answer quality across a wide range of complex tasks. While this approach has proven effective, it inevitably increases substantial inference costs. Previous methods adopting token-level reduction without clear criteria result in poor performance compared to models trained with complete rationale. To address this challenge, we propose a novel sentence-level rationale reduction framework leveraging likelihood-based criteria, *verbosity*, to identify and remove redundant reasoning sentences. Unlike previous approaches, our method leverages *verbosity* to selectively remove redundant reasoning sentences while preserving reasoning capabilities. Our experimental results across various reasoning tasks demonstrate that our method improves performance by an average of 7.71% while reducing token generation by 19.87% compared to model trained with complete reasoning paths.

1 Introduction

Recent advances in Large Language Models (LLMs) have demonstrated remarkable reasoning capabilities comparable to human cognitive abilities (Madaan et al., 2024; Shinn et al., 2024; Kumar et al., 2024). These works demonstrate the capability to solve complex reasoning tasks through explicitly generating extended reasoning paths. The generation of such paths involves producing explicit reasoning units (e.g., tokens, steps) (Yu et al., 2024b), which further enhances model performance through iterative prompting (Wang et al., 2023; Yao et al., 2023). Through this iterative generation of explicit reasoning paths, the model refines and expands its thought processes while incorporating strategic planning and continuous cognitive generation (Xi et al., 2023; Yang et al., 2024).

While the extensive generation of explicit reasoning units leads to improved performance, it inevitably results in higher inference costs and increased latency (Yu et al., 2024b; Wang et al., 2024). Furthermore, fine-tuning LLMs with complete reasoning paths does not necessarily guarantee enhanced performance (Yu et al., 2024b; Deng et al., 2024b; Liu et al., 2024), indicating the necessity for methods that maintain reasoning capabilities while reducing the generation of reasoning units. Despite this apparent requirement, it remains underexplored how to maintain LLM reasoning capabilities while reducing intermediate reasoning paths across diverse tasks.

Previous methods primarily focused on reducing reasoning paths from two distinct perspectives. Some studies have proposed training pipelines that leverage augmented datasets, iteratively generated by foundation LLMs, to fine-tune subsequent LLMs (Yu et al., 2024b; Liu et al., 2024). However, these approaches remain inherently vulnerable due to their significant dependence on the generative capabilities of LLMs.

In response, other works have focused on directly training LLMs without dataset augmentation to reduce explicit reasoning paths. Deng et al. (2023) introduced a knowledge distillation method to distill explicit reasoning into implicit reasoning through token-level hidden states. Deng et al. (2024b) adopted tokens as the reasoning unit for reduction and proposed a heuristic method to internalize explicit intermediate rationale tokens while Hao et al. (2024) compressed complete rationales into the predefined number of hidden states of tokens. However, their methods present a fundamental limitation as they lack sufficient justification for selecting tokens over more linguistically natural units (e.g., sentences) for reduction (Table 1), and they fail to provide principled criteria for the removal process. Moreover, their evaluation has primarily focused on synthetic arithmetic reason-

Findings of the Association for Computational Linguistics: ACL 2025, pages 20769–20784

ing tasks, limiting their applicability to real-world scenarios.

To address these limitations, we propose a novel training method that maintains LLM reasoning performance while systematically reducing redundant reasoning units within the reasoning process. Our method adopts sentences as fundamental reduction units, establishing more linguistically meaningful boundaries compared to token-level approaches. Through empirical analysis, we demonstrate that sentences in early rationale steps can introduce redundancy in the LLM's answer generation process. Inspired by Dong et al. (2023), we introduce the concept of '*verbosity*', a likelihood-based criteria, to identify redundant reasoning sentences. By incorporating *verbosity* identification into the training process, the model excludes redundant reasoning sentences, thereby reducing intermediate token generation. Finally, we demonstrate our method's effectiveness and generalizability across various real-world reasoning datasets, showing our method improves performance by an average of 7.71% while reducing token generation by 19.87% across various LLMs, and through systematic ablation studies, we analyze the contribution of each proposed component.

2 Related Works

2.1 Performance-Cost Tradeoffs in Reasoning Path Generation

Recent research has demonstrated the critical role of generating iterative and refined reasoning paths in enhancing model reasoning capabilities, albeit at increased computational costs (Wang et al., 2023; Yao et al., 2023; Radha et al., 2024; Wang et al., 2024; Madaan et al., 2024; Shinn et al., 2024; Kumar et al., 2024). Self-Consistency (Wang et al., 2023), Tree of Thoughts (ToT) (Yao et al., 2023), and Strategic Chain of Thought (SCoT) (Wang et al., 2024) improve reasoning accuracy through ensemble-based path selection, tree-structured exploration, and adaptive reasoning with an Inner Dialogue Agent, where each approach requires iterative reasoning path generation, resulting in substantial computational overhead.

Concurrently Self-Refine (Madaan et al., 2024) and Reflexion-based framework (Shinn et al., 2024; Kumar et al., 2024) enhance reasoning abilities through iterative feedback-based refinement and reflective path generation, respectively, though both require multiple forward passes through the model. While the iterative generation and refinement of reasoning paths are essential for achieving optimal performance, they inherently increase inference costs and latency. Therefore, it is crucial to investigate methods for efficiently generating these paths.

2.2 Reasoning Path Reduction

To address the computational costs associated with extensive reasoning paths generation, some lines of work (Yu et al., 2024b; Liu et al., 2024) have focused on generating augmented datasets with varying rationale lengths to reduce the generation of reasoning paths. Yu et al. (2024b) employs Self-Consistency to generate multiple reasoning paths for dataset augmentation, then fine-tunes the model to produce direct answers. Liu et al. (2024) developed a heuristic approach to merge reasoning steps and iteratively trained the model to produce shorter reasoning paths, which are then integrated into the progressive training phase. While these approaches demonstrate empirical effectiveness, they exhibit two fundamental limitations: (1) their substantial dependence on LLM generation capabilities introduces inherent instability, and (2) their objective of reducing reasoning paths necessitates the paradoxical creation of datasets requiring extensive reasoning path generation.

To address these weaknesses, another line of work (Deng et al., 2023, 2024b; Hao et al., 2024) has focused on directly training LLMs without augmented datasets. Implicit-CoT (Deng et al., 2023) implements a multi-model framework where an emulator model is trained to predict the teacher's token-level hidden states, and a student model leverages these predicted states to generate answers. ICoT-SI (Deng et al., 2024b) identifies tokens as reduction units, proposing a method to internalize explicit intermediate rationale tokens by progressively eliminating them from the beginning of the reasoning path within the CoT fine-tuning process. However, these methods demonstrate limited generalization across diverse datasets as they have been validated exclusively on simple arithmetic reasoning tasks, such as multiplication problems. This limitation raises concerns about their applicability to real-world scenarios where rationales are expressed in natural language. Furthermore, they do not explore the adoption of principled criteria and linguistically natural units (e.g., sentences). Specifically, the token-level reduction approach may eliminate critical information necessary for answer gen-

eration or distort the semantic information of the sentence. Motivated by these limitations in existing reduction approaches, we examine the redundancy of various sentence positions for potential elimination and propose a novel method with principled criteria that can effectively reduce reasoning paths while maintaining their efficacy.

3 Early Step Rationales are Redundant

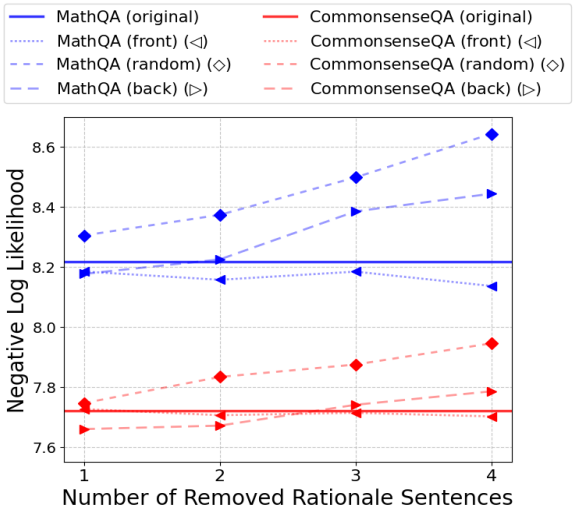

Figure 1: NLL differences across varying sizes of $\{r_i\}$. The 'original' represents the NLL with the complete rationale, while 'front', 'random', and 'back' indicate that $\{r_i\}$ is sampled from the front, random, and back indices of the full index set, respectively.

3.1 Quanitifying the Redundancy

Before delving into the method, we first investigate which positions within the rationale sentences should be selected for reduction. When the likelihood of the answer remains unchanged after removing a sentence from the full rationales, this indicates that the sentence may be redundant in the reasoning process. To quantify the redundancy of a sentence, we compute the negative log-likelihood (NLL) for answer y after sentence reduction as follows:

$$\begin{aligned} \text{NLL} &= -\log p_\theta(y|R', x), \\ \text{where } R' &= R \setminus \{r_i\}_{i \in S},\ S \subseteq I. \end{aligned} \tag{1}$$

Let R denote the complete set of rationale sentences, and I represent the full index set of these sentences. The subset of indices corresponding to sentences selected for reduction is denoted by S, and R' represents the remaining rationale sentences after their removal. $\{r_i\}_{i \in S}$ denotes the sentences selected for reduction. For simplicity, we use $\{r_i\}$ without the subset index notation throughout the rest of the paper.

3.2 Redundancy of Early Reasoning Sentences

We performed a pilot study to empirically demonstrate the redundancy of leading sentences within the rationales by analyzing the NLL of Mistral 7B (Jiang et al., 2023) across diverse reasoning datasets. Specifically, we varied the size of $\{r_i\}$ from 1 to 4 and investigated general patterns of sentence removal using a stochastic approach. To compare different sentence selection configurations, we considered three sampling methods for $\{r_i\}$: **front**, where initial sentences were prioritized; **random**, with uniform probabilities; and **back**, where probabilities progressively increased for later sentences[1]. Additionally, we computed the NLL for complete rationale sentences (i.e., $-\log p_\theta(y|R, x)$) as a baseline to evaluate the impact of reduction. As illustrated in Figure 1, the front (◁) configuration shows only marginal NLL differences relative to complete rationale sentences. In contrast, removing sentences randomly (◇) or from the back (▷) results in higher NLL as the removed sentences increase, highlighting the importance of the selection of a candidate rationale position strategy for removal in the reasoning and answer prediction process (for additional analysis, see Appendix A).

4 Verbosity-Aware Rationale Reduction

Based on these observations, we propose the Verbosity-Aware Rationale Reduction (VARR) framework. In Section 4.1, we introduce the concept of '*verbosity*' as a principled criterion for identifying redundant reasoning sentences. Section 4.2 elaborates on how we integrate *verbosity* into the reduction process during CoT training. In Section 4.3, we extend the *verbosity* term by incorporating incorrect answers to enhance robustness. Finally, Section 4.4 presents the comprehensive VARR framework.

4.1 Verbosity as Principled Criterion

To quantify the redundancy of sentences for potential removal, we introduce the fundamental concept '*verbosity*'. Given an input x, full rationale R, and

[1]For generating R', we assign probabilities $p_k = \frac{N-k+1}{\sum_{i=1}^{N} i}$ (front), $\frac{1}{N}$ (random), and $\frac{k}{\sum_{i=1}^{N} i}$ (back) where k=1,...,N denotes sentence position.

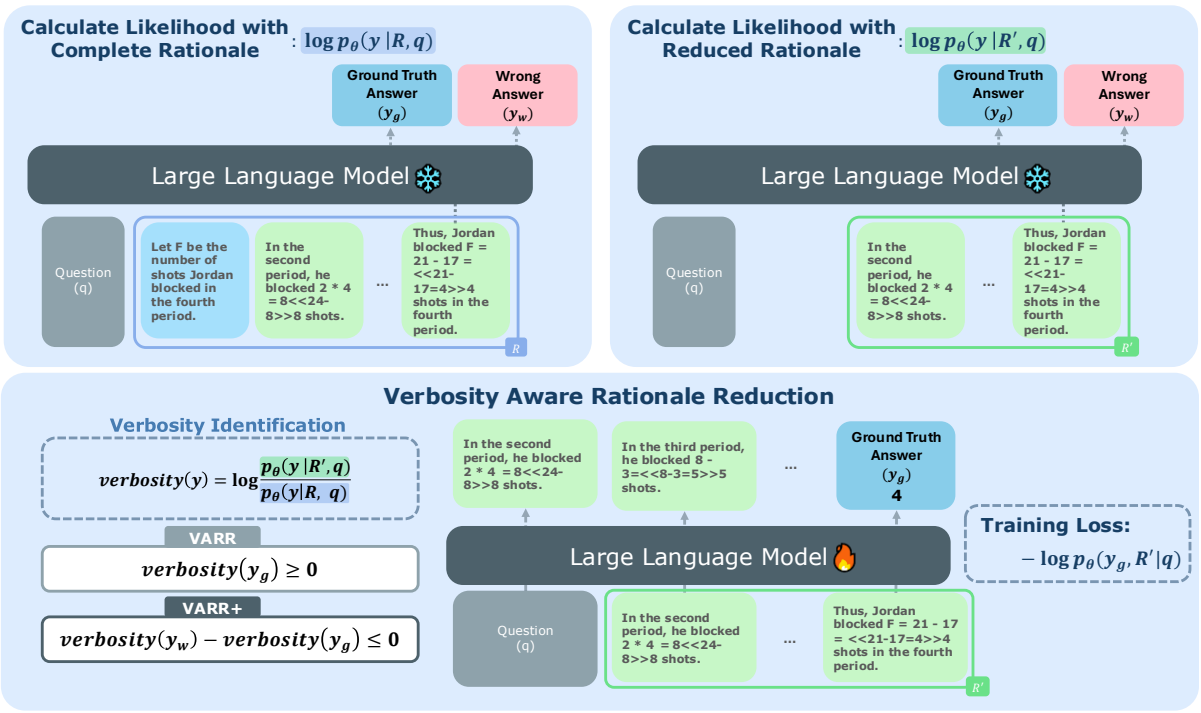

Figure 2: Overview of our VARR/VARR+ framework. Initially, we select a candidate sentence from the beginning of the rationale (Section 3). After selecting the candidate sentence, we evaluate Equations (5) and (9) by calculating $verbosity(y_g)$ and $verbosity(y_w)$. If the candidate sentence meets the verbosity evaluation criteria, it is excluded in subsequent training steps. The model then proceeds with training, where the redundant sentence is excluded from the rationale.

a reduced rationale $R' = \{r_j\}_{j \in I \setminus \{i\}}$, we quantify the *verbosity* of a sentence r_i on y by computing the difference in Kullback–Leibler divergence (KL-divergence) as follows:

$$\begin{aligned} verbosity(y) = & D_{KL}(q(y|x) \parallel p_\theta(y|R,x)) \\ & - D_{KL}(q(y|x) \parallel p_\theta(y|R',x)), \end{aligned} \tag{2}$$

where $q(y|x)$ is the ground truth distribution. The $verbosity(y)$ measures the informational contribution or redundancy of a rationale sentence r_i with respect to answer y. Since $q(y|x)$ is the form of the one-hot vector (i.e., Dirac delta function), we can express the $verbosity(y)$ as the log-likelihood ratio between R and R' as follows:

$$\begin{aligned} verbosity(y_g) &= [H_q(p_\theta(y|R,x)) - H(q(y|x))] \\ &\quad - [H_q(p_\theta(y|R',x)) - H(q(y|x))] \\ &= E_q[-\log p_\theta(y|R,x)] \\ &\quad + E_q[\log p_\theta(y|R',x)] \\ &= \log \left(\frac{p_\theta(y_g|R',x)}{p_\theta(y_g|R,x)} \right), \end{aligned} \tag{3}$$

where y_g denotes the ground truth answer (i.e., $q(y_g|x) = 1$). $H_q(\cdot)$ and $H(\cdot)$ denote the cross-entropy and the entropy, respectively[2]. Intuitvely, a higher value of $verbosity(y_g)$ implies that the likelihood of the model generating the ground truth answer increases after removing r_i, indicating that its removal is beneficial.

4.2 Verbosity Identification in CoT Training

Given an input sequence, CoT training (Nye et al., 2021) aims to train LLMs to generate complete rationale, followed by the ground truth answer:

$$-\log p_\theta(y_g, R|x). \tag{4}$$

During each training step t, we evaluate each sentence r_i within R using the following criterion:

$$verbosity(y_g) \geq 0. \tag{5}$$

Here, we sequentially select r_i starting from the first sentence and construct $R' = \{r_j\}_{j \in I \setminus \{i\}}$, based on our analysis in Section 3, which indicates that early-stage rationales are more likely to be redundant. When the verbosity score is non-positive,

[2]For the sake of explainability, we assume each expression's y is represented by a single token.

it indicates that removing r_i from R would impair the model's performance, thus identifying the sentence as essential and preserving it in the rationale. Detailed training procedures will be described in Section 4.4.

4.3 Contrasting with Wrong Answer

Inspired by the miscalibrated log-likelihoods between accepted and rejected responses in standalone Supervised Fine-Tuning (SFT) for alignment learning (Rafailov et al., 2024; Azar et al., 2024; Hong et al., 2024), we examine whether reduced rationales R' lead to inaccurate answer generation by incorporating a wrong answer y_w.

Instead of employing the correct answer distribution $q(y|x)$ in Equation (2), we initiate our formula with the wrong answer distribution q' (i.e., $1 - q(y|x)$, normalized sum to 1). Through algebraic manipulation, we derive:

$$\begin{aligned} &D_{KL}(q'(y|x) \parallel p_\theta(y|R,x)) \\ &\quad - D_{KL}(q'(y|x) \parallel p_\theta(y|R',x)) \\ &= \left[H_w(p_\theta(y|R,x)) - H(q'(y|x))\right] \\ &\quad - \left[H_w(p_\theta(y|R',x)) - H(q'(y|x))\right], \end{aligned} \tag{6}$$

where $H_w(\cdot)$ denotes the cross-entropy calculated with the wrong answer distribution q'. Due to the impracticality of computing the expectation over the entire space of $V-1$ wrong answers (where V is the vocabulary size), we sample K incorrect answers for the following estimations:

$$\begin{aligned} &E_w[-\log p_\theta(y|R,x)] + E_w[\log p_\theta(y|R',x)] \\ &= E_w\Big[\log \frac{p_\theta(y|R',x)}{p_\theta(y|R,x)}\Big] \\ &\approx \frac{1}{K}\sum_{k=1}^{K} \log \frac{p_\theta(y_w^{(k)}|R',x)}{p_\theta(y_w^{(k)}|R,x)}, \end{aligned} \tag{7}$$

where $\{y_w^{(k)}\}_{k\in[K]}$ is sampled from the in-batch negatives depending on the dataset. Consequently, *verbosity*(y_w) is computed as:

$$\textit{verbosity}(y_w) = \frac{1}{K}\sum_{k=1}^{K} \log \frac{p_\theta(y_w^{(k)}|R',x)}{p_\theta(y_w^{(k)}|R,x)}. \tag{8}$$

Since computational constraints necessitate sampling incorrect answers to calculate *verbosity*(y_w), we evaluate the effectiveness of removal by comparing *verbosity*(y_w) against *verbosity*(y_g) rather than solely using *verbosity*(y_w) as follows:

$$\textit{verbosity}(y_w) - \textit{verbosity}(y_g) \leq 0. \tag{9}$$

When both conditions *verbosity*$(y_g) \geq 0$ and *verbosity*$(y_w) - $ *verbosity*$(y_g) \leq 0$ are satisfied, it indicates that removing r_i from R not only improves the model's performance but also increases its preference for the ground truth answer over incorrect answers, supporting the removal of r_i.

4.4 CoT Training with Rationale Reduction

In our framework, the model is trained using Equation (4) for predefined warm-up stage to inject its reasoning capabilities. Subsequently, at each training step t, we evaluate each sentence r_i sequentially from the first sentence in R, using either Equation (5) alone (denoted as VARR) or the combination of Equations (5) and (9) (denoted as VARR+). Sentences that satisfy these respective criteria are identified for removal and excluded from subsequent training steps. The maximum removable number of sentences at each training step t is determined based on a linear schedule, adopting ICoT-SI (Deng et al., 2024b)'s setting as follows:

$$r(t) = \lfloor N_t \cdot (t/T) \rfloor, \tag{10}$$

where T represents the total number of training steps, N_t is the total number of rationale sentences at step t, and $r(t)$ indicates the maximum number of sentences that can be removed at that step. Note that unlike ICoT-SI, which enforcely removes a predefined number of tokens during training, our method preserves the essential reasoning steps by employing principled removal criteria.

5 Experiments

5.1 Training Configuration

Datasets We conducted experiments across two categories to provide a comprehensive evaluation of VARR's versatility and effectiveness, in contrast to prior research that predominantly focuses on simple arithmetic tasks like multi-digit multiplication (Deng et al., 2023, 2024b). Initially, we evaluate with arithmetic reasoning tasks, including datasets like MathQA (MQA; Amini et al. 2019a) and GSM8K (G8K; Cobbe et al. 2021a). We also examine the performance of our method on commonsense reasoning tasks, employing datasets including CommonsenseQA (CQA; Talmor et al. 2019), TriviaQA (TQA; Joshi et al. 2017), and StrategyQA (SQA; Geva et al. 2021). Note that

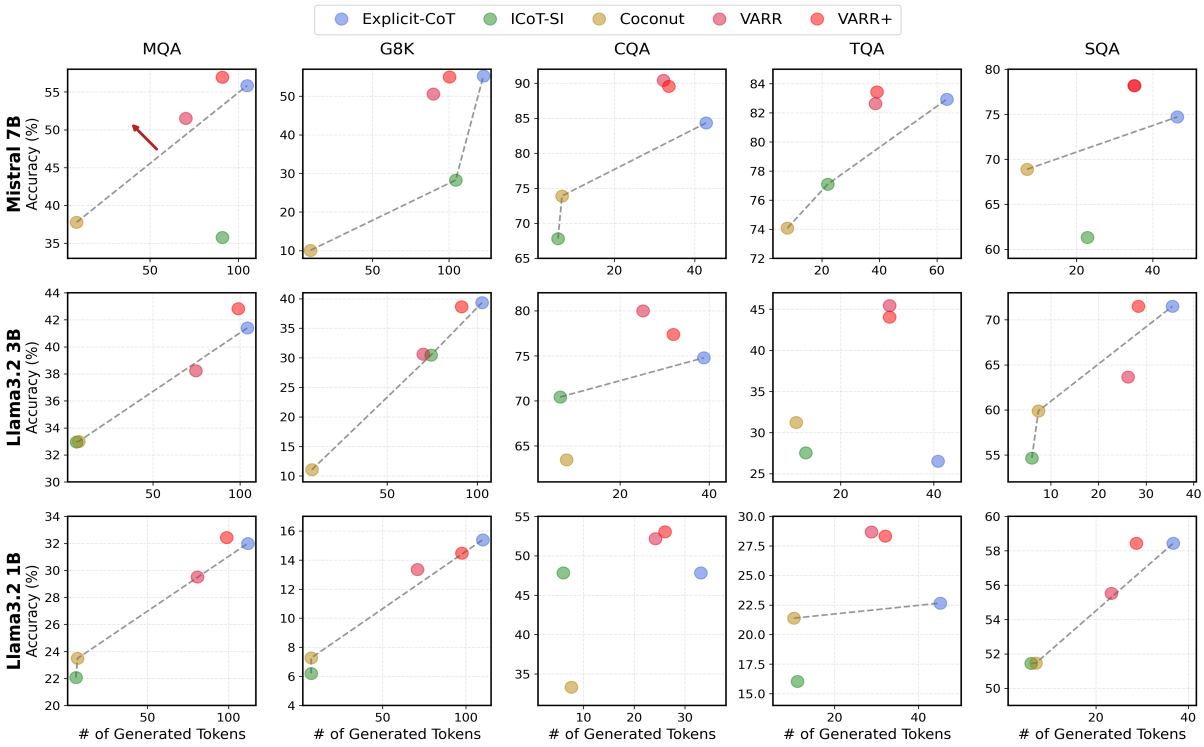

Figure 3: Pareto plot of accuracy versus the number of generated tokens. The gray dotted lines connect the Pareto frontiers of the baselines, and our **VARR (or VARR+) consistently outperforms the pareto frontiers** across all subplots. While ICoT-SI and Coconut substantially trade-off accuracy for efficiency, VARR/VARR+ maintains high accuracy while reducing generated tokens, demonstrating its superior efficiency-performance balance.

unlike previous works (Deng et al., 2023, 2024b), we do not synthesize training data (especially the intermediate steps) to validate the generalizability and applicability of our method.

Models We trained Mistral 7B (Jiang et al., 2023) as our base model for comparisons. We also trained a series of Llama3.2 models (AI@Meta, 2024) scaling from 1B to 3B to demonstrate our method's generalization capabilities.

Implementation Details The warm-up stage is set to 0.1 of the total training steps, and its impact is analyzed in Section 5.5.3. In addition, an optimizer is reinitialized at the beginning of each epoch to stabilize the model training inspired by Deng et al. (2024b)'s setting, with its effects described in Appendix E. All methods are trained for 5 epochs for fair comparison, with detailed implementation of VARR/VARR+ provided in Appendix B.

5.2 Baselines

We compared our method against the following baselines: **Explicit-CoT** (Nye et al., 2021), where the model is finetuned with explicit chain-of-thought reasoning; **ICoT-SI** (Deng et al., 2024b), where the model is fine-tuned using a linear token elimination schedule; and **Coconut** (Hao et al., 2024), where the model is fined-tuned to compress rationales into a predefined number of token hidden states. We excluded Implicit-CoT (Deng et al., 2023) from our evaluation due to its substantial computational demands, specifically requiring three models to be trained simultaneously on a single GPU. Moreover, Deng et al. (2024b) demonstrated that this method achieves a lower performance compared to ICoT-SI. Given that both our baselines and VARR aim to *maintain Explicit-CoT's performance while reducing the number of generated tokens*, we establish Explicit-CoT's performance as our primary baseline for comparison. All baselines were trained on a single A100-80GB GPU, and detailed training configurations for each method are provided in Appendix C.

5.3 Evaluation

We employ two evaluation metrics: First, we evaluate the accuracy of each method in generating the final answer for the respective tasks. Second, we count the generated tokens to evaluate reasoning efficiency while maintaining performance.

5.4 Main results

In Figure 3, we present the results for each reasoning task across different models. VARR/VARR+ achieves comparable or superior performance compared to Explicit-Cot across most datasets while reducing the average token generation. Specifically, VARR+ significantly increases performance by an average of 7.71% across all datasets and models, while improving efficiency by reducing token generation by 19.37% on average.

It is noteworthy that these findings contrast with ICoT-SI, demonstrating that performance can be improved while reducing the number of generated tokens. This suggests that existing reasoning data contains unnecessary reasoning sentences that may harm performance. Furthermore, effective reasoning can be achieved by selectively removing sentences based on appropriate criteria.

However, the baselines exhibit performance degradation compared to Explicit-CoT. For ICoT-SI, we observe an average performance decline of 21.98%, while Coconut shows a degradation of 25.20%, demonstrating their imbalanced trade-off in efficiency. These results suggest that heuristic reasoning reduction approaches do not effectively induce implicit reasoning within the model as discussed by ICoT-SI and Coconut. This indicates that identifying and retaining appropriate reasoning units through principled criteria is crucial for maintaining performance in practical applications. A more detailed discussion of the impact of the choice of reduction unit and criterion is provided in Section 5.5.1.

Furthermore, qualitative analysis (refer to Appendix J) confirms that ICoT-SI and Coconut fail to generate valid reasoning paths for answer generation, while VARR+ produces concise, yet effective reasoning paths that lead to correct answers. Additionally, the incorporation of incorrect answers in VARR+ resulted in performance improvements across most datasets. While VARR alone enhanced generation efficiency, VARR+ effectively preserves rationales that help calibrate the probability distribution between correct and incorrect answers, contributing to improved training robustness and stability.

5.5 Ablation Studies

In this section, we conduct ablation studies to empirically validate our method. All experiments are implemented using Mistral 7B due to its higher base capacity compared to other models.

5.5.1 Identifying Appropriate Units for Removal

	MQA	G8K	CQA	TQA	SQA
Exp-CoT	55.84 (105.02)	**55.26** (122.54)	84.33 (42.84)	82.94 (63.46)	74.70 (46.47)
ICoT-SI	35.84 (113.55)	28.27 (104.41)	67.82 (6.0)	77.09 (22.03)	61.33 (22.86)
VARR-Tok	46.79 (90.39)	47.53 (94.16)	82.60 (25.07)	67.14 (38.10)	71.22 (31.92)
VARR-Sent	**56.95** (91.04)	54.98 (100.38)	**89.56** (33.55)	**83.45** (39.17)	**78.19** (35.12)

Table 1: Analysis across various reasoning reduction units and the application of principled criteria. Each row presents accuracy in the first line, with average generated tokens shown in parentheses in the second line.

In this section, we empirically investigate the necessity of our criteria and demonstrate why sentences are more effective than tokens as reasoning reduction units. While ICoT-SI removes tokens without specific criteria, we first apply VARR+ at the token level (denoted as VARR-Tok) to assess whether tokens can serve as effective reduction units when combined with our criteria. As shown in Table 1, VARR+ applied at the token level achieves an average performance gain of 24.74% compared to ICoT-SI, demonstrating that our principled criteria contribute to robust performance. Furthermore, expanding the reduction units from tokens to sentences (denoted as VARR-Sent) yields an additional performance gain of 15.98% over VARR-Tok. These findings highlight that sentences provide natural and effective boundaries for the reduction process.

5.5.2 Analyzing the Impact of Sentence Position on Removal Efficacy

In Section 3, we demonstrated that gradually removing sentences from random and back positions can degrade model performance. To further explore this finding and assess the robustness of removing sentences from the front position, we conducted experiments with unguided random sentence removal (denoted as No Rule) and applied VARR+ with random position and reverse sentence order (denoted as Random and Back, respectively). As shown in Table 2, unguided random sentence removal resulted in a 25.30% decrease in performance relative to our method, highlighting the critical role of the verbosity evaluation even after selecting sen-

	MQA	G8K	CQA	TQA	SQA
Exp-CoT	55.84 (105.02)	**55.26** (122.54)	84.33 (42.84)	82.94 (63.46)	74.70 (46.47)
No Rule	35.06 (59.92)	25.85 (63.56)	74.78 (23.56)	74.14 (6.84)	72.38 (23.38)
Random	55.34 (99.13)	52.31 (115.59)	83.47 (49.19)	79.62 (44.98)	73.83 (50.51)
Back	49.71 (92.64)	48.52 (103.25)	85.21 (19.86)	70.0 (36.22)	74.41 (18.17)
Front	**56.95** (91.04)	54.98 (100.38)	**89.56** (33.55)	**83.45** (39.17)	**78.19** (35.12)

Table 2: Performance across 5 different reasoning tasks, evaluated with different sentence position removal. Each row presents accuracy in the first line, with average generated tokens shown in parentheses in the second line.

tences as units of reduction. Furthermore, Random and Back strategies exhibited an average 7.50% performance degradation relative to our method. These results further support our observation that earlier sentences in the reasoning path tend to contain more redundancy, and their prioritized removal effectively balances rationale reduction while maintaining reasoning performance.

	G8K	SQA	TQA
Exp-CoT	**55.84** (122.54)	74.70 (46.47)	82.94 (63.46)
RAND-SENT REMOVAL	25.85 (63.56)	72.38 (23.38)	74.14 (6.84)
FRONT-SENT REMOVAL (1)	38.36 (87.43)	72.96 (24.34)	79.74 (22.73)
FRONT-SENT REMOVAL (2)	23.50 (47.64)	73.83 (9.97)	76.79 (8.32)
VARR+	54.98 (100.38)	**78.19** (35.12)	**83.45** (39.17)

Table 3: Performance across different front sentence removal strategies. RAND-SENT-REMOVAL refers to the Random strategy, while $(\cdot)$ in the FRONT-SENT REMOVAL indicates the number of enforcedly removed front sentences. Each row presents accuracy in the first line, with average generated tokens shown in parentheses in the second line.

To further validate whether VARR enhances robustness compared to enforced removal of front sentences, we conducted additional experiments where the first and second sentences were enforced to be removed during the initial two training epochs. As shown in Table 3, enforced removal of front sentences leads to substantial performance degradation, while VARR demonstrates its contribution to preserving robustness.

5.5.3 Varying the Warm-up Ratio

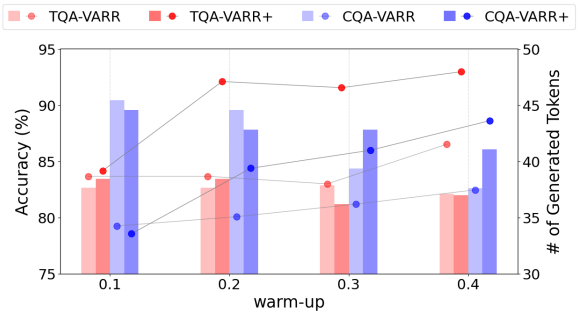

Figure 4: Accuracy (barplot) and the average generated token (marker) across various warm-up stages on TriviaQA and CommomsenseQA.

We evaluated our method against various warm-up stages on the TriviaQA and CommonsenseQA datasets. As the duration of the warm-up stages increases, the model becomes more fitted to the non-reduction dataset, which inhibits VARR's ability to eliminate redundant sentences from the reasoning path. Consequently, as illustrated in Figure 4, longer warm-up periods result in an increase in generated tokens and a decrease in accuracy. These results suggest that 0.1 training steps provide sufficient time to inject reasoning abilities while enabling the systematic reduction of redundant reasoning sentences during the learning process. The zero warm-up configuration demonstrates comparable performance, though 0.1 warm-up steps are slightly better (detailed in Appendix F). Therefore, we set 0.1 as the default setting for the warm-up steps.

5.5.4 Removal Ratio Analysis

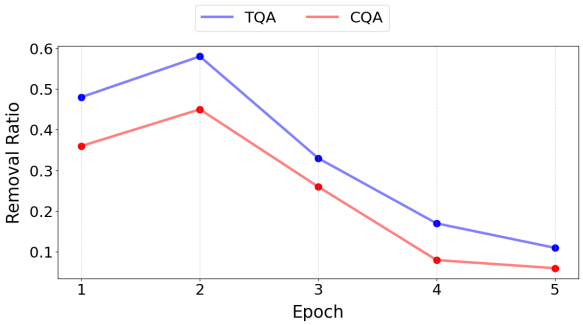

Figure 5: Removal ratio of redundant sentences during training. The y-axis shows the removal ratio, calculated as the number of removed sentences divided by the maximum potential removal sentences ($size(\{r_i\})/r(t)$). The x-axis represents training epochs.

In Figure 5, we analyze the actual amount of

rationale sentences removed during training. We examine it by calculating the removal ratio, the proportion of actual removed sentences to the maximum potential removal sentences $r(t)$. Our analysis indicates that not all sentences designated for maximum removal range are always eliminated during the training process. Notably, a significant proportion of redundant sentences are removed in the early stages of training, with fewer sentences being removed as the model progresses through the middle to later stages, thereby stabilizing its training. A similar trend is observed across other datasets, as detailed in Appendix G.

6 Conclusion

In this work, we propose the sentence-level rationale reduction framework VARR and empirically demonstrate that models trained with non-redundant rationales achieve enhanced efficiency. We address the lack of principled criteria for identifying redundant sentences during training by developing a reduction framework that not only preserves the model's reasoning capabilities but also reduces the likelihood of generating incorrect answers. Our experiments show that VARR can efficiently handle a diverse range of tasks with fewer generated tokens, without sacrificing its accuracy. This work contributes novel insights to rationale reduction research, contributing to the efficient reasoning elicitation in language models.

Limitations

While our work provides novel insights into rationale reduction research, our experiments were primarily conducted using a relatively small large language model and limited batch size, constrained by computational costs (i.e., a single A100-80GB GPU). Additionally, for the same reasons, it was not feasible to test the model with datasets featuring long sequences in both queries and rationales (Reddy et al., 2024; Yu et al., 2024a). Nevertheless, given the systematic design principles underlying the VARR/VARR+ frameworks, we believe their effectiveness would extend to larger-scale implementations. Furthermore, we reserve the application of VARR/VARR+ in iterative reasoning path generation and refinement/reflexion-based evaluation discussed in Section 2.1 for future work.

Ethical Considerations

Our work explores how LLMs can maintain their reasoning performance while improving efficiency. To this end, we conducted verbosity-aware rationale reduction (i.e., reasoning sentence pruning)-based CoT fine-tuning, requiring computational resources comparable to standard CoT fine-tuning. Additionally, we used only open-source LLMs and publicly available reasoning datasets with minimal preprocessing using `gpt4o-mini`'s (OpenAI, 2024) API. Therefore, we do not anticipate significant ethical issues arising from our work. On the contrary, we believe future works could leverage our analysis to reduce computational overhead in CoT inference settings.

Acknowledgements

We sincerely appreciate to Keonwoo Kim and Hyowon Cho for their sharp feedback during writing, Sangyeop Kim, Yukyung Lee, and Minjin Jeon for their continuous encouragement and inspiration during our study sessions, and Eunbi Choi for her invaluable help during the rebuttal process. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2019-II191906, Artificial Intelligence Graduate School Program(POSTECH)), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2018-0-00584, (SW starlab) Development of Decision Support System Software based on Next-Generation

Machine Learning), National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00335873), and National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00217286).

References

LlamaTeam AI@Meta. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019a. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019b. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR.

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

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

Yihe Deng, Weitong Zhang, Zixiang Chen, and Quanquan Gu. 2024a. Rephrase and respond: Let large language models ask better questions for themselves. *Preprint*, arXiv:2311.04205.

Yuntian Deng, Yejin Choi, and Stuart Shieber. 2024b. From explicit cot to implicit cot: Learning to internalize cot step by step. *Preprint*, arXiv:2405.14838.

Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. 2023. Implicit chain of thought reasoning via knowledge distillation. *Preprint*, arXiv:2311.01460.

Qingxiu Dong, Jingjing Xu, Lingpeng Kong, Zhifang Sui, and Lei Li. 2023. Statistical knowledge assessment for large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

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*.

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

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Orpo: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11170–11189.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Seungone Kim, Se June Joo, Doyoung Kim, Joel Jang, Seonghyeon Ye, Jamin Shin, and Minjoon Seo. 2023. The cot collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. 2024. Training language models to self-correct via reinforcement learning. *Preprint*, arXiv:2409.12917.

Tengxiao Liu, Qipeng Guo, Xiangkun Hu, Cheng Jiayang, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2024. Can language models learn to skip steps? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.

OpenAI. 2024. Gpt-4o mini: advancing cost-efficient intelligence.

Santosh Kumar Radha, Yasamin Nouri Jelyani, Ara Ghukasyan, and Oktay Goktas. 2024. Iteration of thought: Leveraging inner dialogue for autonomous large language model reasoning. *Preprint*, arXiv:2409.12618.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Varshini Reddy, Rik Koncel-Kedziorski, Viet Dac Lai, Michael Krumdick, Charles Lovering, and Chris Tanner. 2024. Docfinqa: A long-context financial reasoning dataset. *arXiv preprint arXiv:2401.06915*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Damien Sileo. 2024. tasksource: A large collection of NLP tasks with a structured dataset preprocessing framework. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15655–15684, Torino, Italia. ELRA and ICCL.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North*, page 4149. Association for Computational Linguistics.

Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. 2024. Openmathinstruct-2: Accelerating ai for math with massive open-source instruction data. *arXiv preprint arXiv:2410.01560*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Yu Wang, Shiwan Zhao, Zhihu Wang, Heyuan Huang, Ming Fan, Yubo Zhang, Zhixing Wang, Haijun Wang, and Ting Liu. 2024. Strategic chain-of-thought: Guiding accurate reasoning in llms through strategy elicitation. *Preprint*, arXiv:2409.03271.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wensen Cheng, Qi Zhang, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huang, and Tao Gui. 2023. The rise and potential of large language model based agents: A survey. *Preprint*, arXiv:2309.07864.

Ling Yang, Zhaochen Yu, Tianjun Zhang, Shiyi Cao, Minkai Xu, Wentao Zhang, Joseph E. Gonzalez, and Bin CUI. 2024. Buffer of thoughts: Thought-augmented reasoning with large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Zhiyuan Zeng, Xiaonan Li, Junqi Dai, Qinyuan Cheng, Xuan-Jing Huang, and Xipeng Qiu. 2024. Reasoning in flux: Enhancing large language models reasoning through uncertainty-aware adaptive guidance. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2401–2416.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024a. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*.

Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. 2024b. Distilling system 2 into system 1. *arXiv preprint arXiv:2407.06023*.

A Additional Datasets Analysis

As shown in Section 3, the NLL tends to rise with increasing size of $\{r_i\}$ in random and back configurations, compared to the front configuration, as illustrated in Figure 6, this trend is consistent across both the GSM8K and StrategyQA datasets.

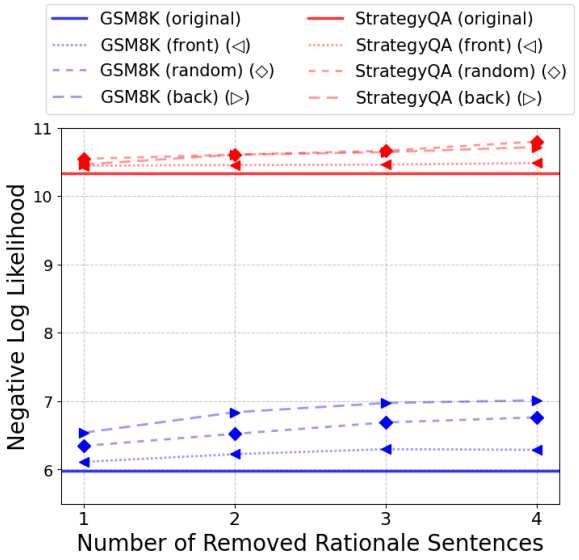

Figure 6: NLL differences across varying sizes of $\{r_i\}$. The 'original' represents the NLL for the full rationale, while 'front', 'random', and 'back' indicate that r_i are sampled from the front, random, and back indices of the full index set, respectively.

B VARR Implementation

The detailed implementation of VARR+ is outlined in Algorithm 1. After the warm-up stages, each datum in the current batch is evaluated using the verbosity equation—specifically, using only Equation 5 in VARR and both Equations 5 and 9 in VARR+. As mentioned in Section 3 and 4, the process assesses the redundancy of each sentence in data starting from the first index during every single epoch. Therefore, all data can potentially shorten the length of the rationale progressively.

C Additional Experimental Details

For all experiments, we employ the AdamW optimizer (Loshchilov and Hutter, 2019), configured with a weight decay of 0.005. For the Mistral 7B model, we utilize an effective batch size of 12 with gradient accumulation set to 3, while the smaller Llama3.2 models use an effective batch size of 15. For Coconut (Hao et al., 2024), we set the max_latent_stage to 5 while maintaining all other

Algorithm 1 Training Procedure of VARR+

$\mathcal{D}$: Training dataset
$\mathcal{B}$: Training batch
E: Total training epochs
S: Total training steps per epoch
$T = E \times S$: Total training steps
$T_{warmup} = 0.1 \times T$: Warm-up steps
R_B: A buffer to store removed sentences
$r(t)$: Maximum number of removable sentences at t
N_t: Number of rationale sentences at t
θ: Trained model parameters
for epoch = 1 to E **do**
  for step = 1 to S **do**
    Sample training batch $\mathcal{B}$ from $\mathcal{D}$
    $t = (epoch - 1) \times S + step$
    if $t \leq T_{warmup}$ **then**
      pass
    end if
    for each $d \in \mathcal{B}$ **do**
      $R_B \leftarrow \{\}$
      for i = 1 to N_t **do**
        if r_i satisfies Equations 5 and 9 **then**
          remove r_i from d
          Add r_i to R_B
          if $|R_B| \geq r(t)$ **then**
            break
          end if
        end if
      end for
    end for
    Forward pass
    Backward pass and update θ
  end for
  Reinitialize optimizer
end for

hyperparameters as their default repository values unless otherwise mentioned. A constant learning rate of 5×10^{-6} is applied across all datasets, with bfloat16 precision. For Multiple-Choice and True/False tasks, complete sets of non-correct labels are employed to configure in-batch negatives to enhance the stability of the verbosity evaluation. To ensure a fair comparison, all baselines and methods are trained on a single A100 GPU with 80GB of memory for up to 5 epochs or 36 hours, whichever is reached first and experimented with single run evaluation (note that our setting is different from prompting/inference only setting). Regarding licensing, Mistral 7B is licensed under Apache License, Version 2.0, while Llama3.2 is governed by the Llama 3.2 Community License.

D Details and Statistics of Datasets

For our experimental analysis, we carefully selected a diverse set of five datasets used in prior works (Deng et al., 2023, 2024a; Liu et al., 2024; Yu et al., 2024b; Yin et al., 2024). To ensure explicit sentence boundaries, all datasets were prepro-

Dataset	Reasoning Task	Source	Answer Format	# TRAIN.	# VALID.	# TEST.
GSM8K (Cobbe et al., 2021a)	Arithmetic	Cobbe et al. (2021b)	Number	7000	473	1319
MathQA (Amini et al., 2019a)	Arithmetic	Amini et al. (2019b)	Multi-choice	29837	4475	2985
TriviaQA (Joshi et al., 2017)	Commonsense	Kim et al. (2023)	Natural Language	8844	552	1659
CommonsenseQA (Talmor et al., 2019)	Multi-choice	Kim et al. (2023)	Multi-choice	609	38	115
StrategyQA (Geva et al., 2021)	Commonsense	Sileo (2024)	T/F	1832	114	344

Table 4: Comprehensive statistics of the datasets used in our experiments are provided. GSM8K and MathQA are sourced directly from their original datasets, while the remaining datasets were obtained from Kim et al. (2023) and Sileo (2024) to access complete reasoning paths. GSM8K, StrategyQA, and CommonsenseQA are licensed under the MIT License, whereas MathQA and TriviaQA are distributed under the Apache License, Version 2.0.

cessed using `gpt-4o-mini` (OpenAI, 2024) to establish clear sentence demarcation (e.g., 'He bikes 20*2=«20*2=40»40 miles each day for work So he bikes 40*5=«40*5=200»200 miles for work' becomes 'He bikes 202=«202=40»40 miles each day for work. So he bikes 405=«405=200»200 miles for work'). Table 4 comprehensively outlines each dataset, including its source and the size of the training, validation, and test samples.

	MQA	G8K	CQA	TQA	SQA
w/o reinit	48.70 (92.86)	45.18 (115.91)	87.82 (32.95)	83.72 (38.27)	76.74 (33.89)
w/ reinit	56.95 (91.04)	54.98 (100.38)	89.56 (33.55)	83.45 (39.17)	78.19 (35.12)

Table 5: Performance comparison before and after the reinitializing optimizer in every training epoch. Each row presents accuracy in the first line, with average generated tokens shown in parentheses in the second line.

E Reinitializing the Optimizer

We reinitialized the optimizer after each training epoch to stabilize training, inspired by Deng et al. (2024b). Our implementation uses the AdamW optimizer (Loshchilov and Hutter, 2019), where the first and second moments are gradually updated based on current gradients. Consequently, when VARR reduces rationales for certain data points between epochs, the training process could become unstable. As shown in table 6, the average 19.32% performance improvement achieved through optimizer reinitialization extends the findings of Deng et al. (2024b) beyond simple tasks (e.g., multiplication; synthesized dataset) to demonstrate effectiveness across diverse datasets with complex semantic and syntactic reasoning structures.

F Zero Warm-up Stage

To validate whether the warm-up stage sensitivity could limit applicability, we conducted additional

warm-up	0.0	0.1	0.2	0.3	0.4
TQA	82.01 (40.11)	83.72 (38.27)	83.45 (47.10)	81.19 (46.55)	81.97 (47.97)
CQA	90.34 (32.81)	89.56 (35.07)	87.82 (39.38)	87.82 (40.98)	86.08 (43.61)

Table 6: Accuracy and average generated tokens across the various warm-up stages.

experiments extending warm-up stages to zero. As shown in Table, the zero warm-up stage shows comparable effectiveness and efficiency to the 0.1 warm-up stage, demonstrating the robustness of our method across different application scenarios. These findings suggest a zero-warm stage can be a good option for various application scenarios, but starting with a 0.1 warm-up stage can boost its stability.

G Removal Ratio Analysis on additional datasets.

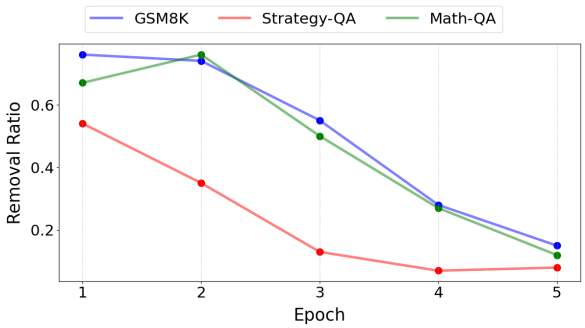

Figure 7: Additional analysis of the Removal ratio on other datasets. The Removal ratio, indicated on the y-axis, is calculated by dividing the number of sentences actually removed by the maximum potential removal ratio (i.e., $size(\{r_i\})/r(t)$). The x-axis represents the index of the epochs.

In the GSM8K, StrategyQA, and MathQA datasets, similar trends are observed as in Section 5.5.4. Additionally, we observe that our method reduces fewer sentences in later train-

ing stages, demonstrating that *verbosity* preserves essential reasoning sentences even when larger $r(t)$ encourages substantial reduction. Notably, in StrategyQA, the Removal ratio converges more rapidly than in other datasets. This is attributed to the dataset's characteristics, where the sentences within the rationale predominantly list simple information rather than forming complex, interrelated structures. Consequently, redundant sentences are rapidly removed, leading to a rapid decrease in further removal activities.

H Exploration on Complex Task

	Exp-CoT	VARR+
Acc	9.01	10.27
Gen Tok	300.17	252.00

Table 7: Performance comparison between Exp-CoT and VARR+ on MATH datasets.

To further validate that our method remains effective on complex tasks, we conducted additional experiments with the MATH (Hendrycks et al., 2021) dataset. Considering computational constraints, we selected 500 samples from the math subset of OpenMathInstruct-2 (Toshniwal et al., 2024) for the training set and evaluated on the official test set. As described in Table 8, our results on MATH further demonstrate the versatility and future applicability of VARR+ on more complex tasks.

I Training Time Analysis

Dataset	GSM8K (7K)	TQA (2K)
Training Time (↑)	0.18	0.30
Token Efficiency (↓)	0.81	0.61
Accuracy (↑)	0.98	1.01

Table 8: Training time analysis on GSM8K and TQA. All scores are calculated as the value for Explicit-CoT divided by the value for VARR+ (i.e., Explicit-CoT/VARR+).

As VARR computes the likelihood of each sentence to ensure robust performance, training time inevitably increases as dataset size grows. We therefore quantitatively measure its efficiency on GSM8K and TQA relative to Explicit-CoT. Although there is a trade-off in training time, VARR+ improves token efficiency while preserving accuracy. Since training occurs only once while the improved efficiency benefits persist continuously afterward in test-time, we believe the additional training cost is justified.

J Qualitative Analysis

Table 9 illustrates the generation outputs for each method on the GSM8K test set. While Explicit-CoT generates a complete reasoning path, its redundant generation of 'needs a 3 piece place setting' introduces hallucinations into the reasoning process, ultimately leading to an incorrect answer. ICoT-SI's token-level elimination of reasoning paths, on the other hand, results in generations that lack sufficient reasoning capacity. Additionally, Coconut's predefined hidden-states fail to compress the complete reasoning path in its token-level compression approach. In contrast, VARR+ demonstrates efficient generation by concisely producing '$15.00' in a single sentence, achieving both effectiveness and efficiency in its output. Additional qualitative examples on the MathQA dataset are outlined in Table 10.

Question
Avery needs to buy a 3 piece place setting (dinner & salad plate and a bowl) for her holiday dinner. She's having 12 people over for dinner. If the dinner plates cost $6.00 each and bowls each cost $5.00 and the salad plates cost $4.00, how much will she spend on place settings?
Explicit-CoT
She's having 12 people over and needs a 3 piece place setting for each so she needs 12*3 = 36 place settings. The dinner plates cost $6.00 each and she needs 36 of them so that comes to 6*36 = $216.00. The bowl costs $5.00 each and she needs 36 of them so that comes to 5*36 = $180.00. The salad plates cost $4.00 each and she needs 36 of them so that comes to 4*36 = $144.00. All total, the place settings will cost $216 for dinner plates, $180 for bowls and $144 for salad plates for a total of 216+180+144 = $540.00 #### 540
ICoT-SI
each so 12*6 = $72.00 She's buying 3 pieces of place settings (dinner plates, salad plates and bowls) for $72.00 each so 3*72 = $216.00 #### 216
Coconut
<|start-latent|><|latent|><|latent|><|latent|><|end-latent|> #### 4.00
VARR+
She needs 12 place settings and each setting costs $15.00 so she will spend 12*15 = $180.00 #### 180
Ground Truth Answer
180

Table 9: Qualitative Example: Generation outputs on GSM8K test set.

Question
The cost price of 20 articles is the same as the selling price of x articles. if the profit is 25%, find out the value of x. a) 13, b) 14, c) 15, d) 16, e) 17
Explicit-CoT
explanation : let the cost price 1 article = rs . 1 cost price of x articles = rs . x selling price of x articles = 125% of x = rs . (125 / 100 x) = rs . (25 / 20 x) = rs . 5 / 4 x = > x = 20 cost price of 20 articles = rs . 20 selling price of 20 articles = rs . (25 / 4 $\times$ 20) = rs . (25 / 4 $\times$ 20) / 20 = rs . 1 = > x = 20 answer : option a #### a
ICoT-SI
of x articles = x s.p of x articles = rs . 20 gain = rs . (20 - x) gain % = 25 % of x = rs . (20 - x) / x $\times$ 100 = 25 = > 20 - x = 5 x = > 4 x = 20 - x = > 5 x = 20 - x = > 20 = 5 x = > 200 = 5 x^2 = > 400 = 5 x^2 = > 400 / 5 = x^2 = > x^2 = 80 = > x = 10 or 10 . c #### c
Coconut
<|start-latent|><|latent|><|latent|><|latent|><|end-latent|> #### c
VARR+
explanation : let the cost price of x articles = x s.p of x articles = 20 20 = 1.25 x = > x = 16 option d. #### d
Ground Truth Answer
d

Table 10: Qualitative Example: Generation outputs on MathQA test set.