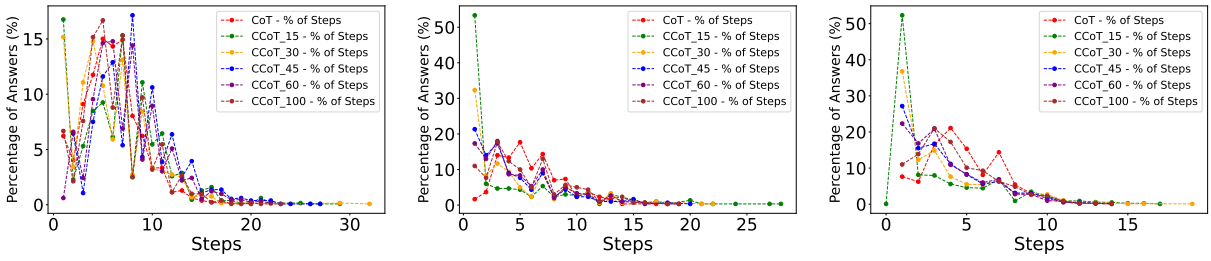
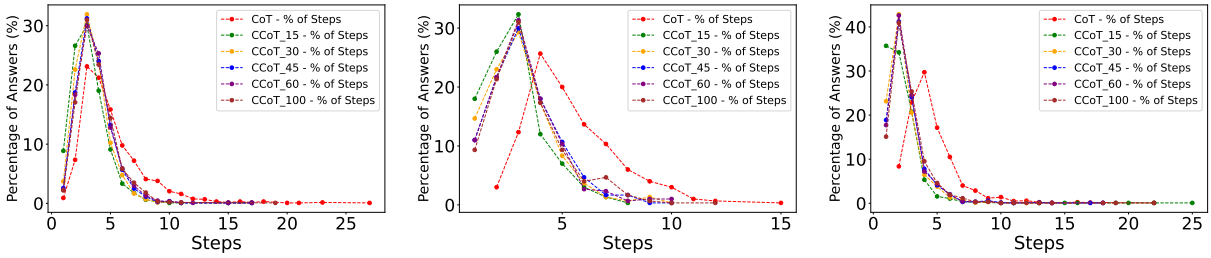


(a) GSM8K - Llama2-70b (b) SVAMP - Llama2-70b (c) ASDIV - Llama2-70b



(d) GSM8K - Falcon-40b (e) Svamp - Falcon-40b (f) ASDIV - Falcon-40b

Figure 4: Percentage of number of generated answers as per 'no. of reasoning steps' of Llama2-70b (top row) and with Falcon-40b (bottom row) on the GSM8K (a), SVAMP (b), and ASDIV (c) test datasets.

datasets (Figures 4a and 4b) reveals that the quartile ranges of the answer distributions for CCoT and CoT answers are very similar. Specifically, in both datasets, the interquartile range spans approximately 4 to 12 steps (Q1 to Q3) for SVAMP and around 5 to 14 steps for GSM8K. This suggests comparable step counts across CCoT and CoT answers, within such ranges of reasoning steps.

To evaluate the conciseness of these reasoning steps more effectively, we present redundancy scores in Figure 5. This figure highlights the differences in redundancy between CCoT and CoT answers, categorized by their number of steps. Notably, when focusing on step intervals within the interquartile range (Q1 to Q3, where most answers fall), which are highlighted in grey in the plots, redundancy scores are consistently higher for CoT than for CCoT. This demonstrates that CCoT achieves improved syntactic conciseness compared to CoT, even when the number of steps is similar.

To better quantify the reduction in redundancy achieved by CCoT, we define the *Mean Redundancy Reduction (MRR)* across single-step redundancies RMS_i for all reasoning steps i, and the *Overall Redundancy Reduction (ORR)* as:

$$\text{MRR} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\text{CoT RMS}_i - \text{CCoT RMS}_i}{\text{CoT RMS}_i} \times 100\right)$$

$$\text{ORR} = \frac{\text{CoT RMS} - \text{CCoT RMS}}{\text{CoT RMS}} \times 100$$

Table 3 presents the average ORR and MRR calculated for the SVAMP and GSM8K datasets, focusing on steps within the interquartile range (Q1 to Q3). The results indicate that CCoT consistently reduces redundancy. For SVAMP, the redundancy reduction ranges from 19.77% to 22.72% (ORR), while for GSM8K, it ranges from 12.64% to 24.74% (ORR).

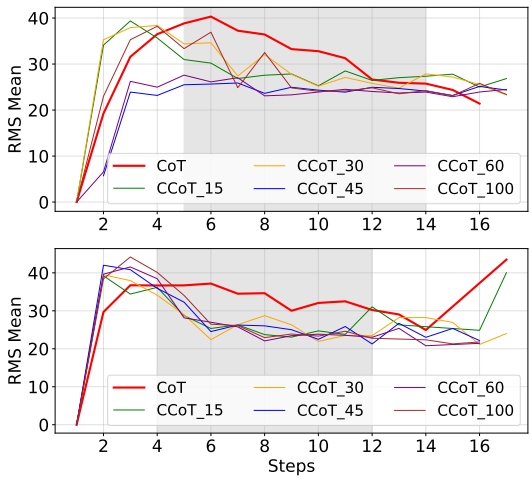


Figure 5: RMS score (lower values indicate better conciseness) of Llama2-70b on the GSM8K (top) and SVAMP (bottom) datasets. The grey area highlights the portion of the answer distribution between the Q1 and Q3 quartiles.

Method	Dataset	CCoT15	CCoT30	CCoT45	CCoT60	CCoT100
MRR	SVAMP	20.02	22.50	21.48	22.92	20.14
	GSM8K	13.58	11.65	22.81	23.23	16.33
ORR	SVAMP	20.22	22.57	21.29	22.72	19.77
	GSM8K	15.34	12.64	24.62	24.74	16.81

Table 3: Mean and Overall Redundancy Reduction for SVAMP and GSM8K Datasets

Information Flow Evaluation. To demonstrate an improved level of conciseness also from a semantic perspective, we analyze the information flow. Specifically, the median number of steps in the answer distributions for CoT and CCoT (Figures 4a and 4b) is approximately 8 for both the SVAMP and GSM8K datasets. Thus, focusing on answers with a total of 8 steps, we present in Tables 4 the *Information Flow* between consecutive steps $i \rightarrow j$ for Llama2-70b on the GSM8K (top) and SVAMP (bottom) datasets.

In Table 4, CCoT-15 exhibits the largest reductions across all steps, for instance, ranging from 26% to 41% compared to CoT scores for GSM8K, while CCoT-100 shows the smallest reductions (4% to 20%), preserving more redundant semantic information. Generally, the middle steps show the largest differences, especially for aggressive variations like CCoT-15 and CCoT-45, whereas early and late steps retain more information flow. In summary, a lower information flow indicates that the model effectively retains, step by step, only the logically necessary information required to arrive at a correct answer.

GSM8K - Llama2-70b

Steps	CoT	CCoT-15	CCoT-30	CCoT-45	CCoT-60	CCoT-100
1 ⇒ 2	0.5287	0.3417	0.39	0.49	0.49	0.43
2 ⇒ 3	0.59	0.39	0.45	0.31	0.32	0.53
3 ⇒ 4	0.56	0.37	0.43	0.36	0.37	0.47
4 ⇒ 5	0.55	0.38	0.41	0.36	0.36	0.46
5 ⇒ 6	0.56	0.39	0.42	0.36	0.37	0.51
6 ⇒ 7	0.56	0.42	0.45	0.37	0.37	0.54
7 ⇒ 8	0.52	0.42	0.47	0.57	0.56	0.50

SVAMP - Llama2-70b

Steps	CoT	CCoT-15	CCoT-30	CCoT-45	CCoT-60	CCoT-100
1 ⇒ 2	0.54	0.32	0.41	0.41	0.30	0.32
2 ⇒ 3	0.52	0.35	0.47	0.43	0.35	0.35
3 ⇒ 4	0.51	0.34	0.46	0.40	0.30	0.37
4 ⇒ 5	0.51	0.37	0.43	0.39	0.30	0.30
5 ⇒ 6	0.51	0.38	0.41	0.44	0.36	0.37
6 ⇒ 7	0.50	0.41	0.48	0.40	0.36	0.40
7 ⇒ 8	0.47	0.35	0.43	0.37	0.34	0.50

Table 4: Information Flow Mean Values comparison for GSM8K (top) and SVAMP (bottom) across answers with 8 steps. Better information flow indicates lower semantic conciseness between steps (highlighted with blue-like colors in the tables).

7.5 Ability to control the output length

The previous experiments looked at how CCoT strategies can affect the accuracy and generation time in the average. However, despite the discussed benefits, it is also crucial to understand how CCoT prompting can effectively limit the output length for each addressed sample (RQ3). This can be useful for better tuning the length parameter in the CCoT prompt or identifying the conditions in which the proposed prompting strategy fails to compress the output. To evaluate the ability of an LLM to produce concise answers in response to a given prompting, we analyzed in Figure 6 the output length under different CCoT length constraints.

Figure 6 shows the statistics on the length of the answers provided by addressed models with the GSM8K test set. Each box plot represents the output lengths between the 5th and the 95th percentiles of all tested samples, the blue line represents the provided CCoT length constraint, the red line denotes the median, while the greed dot the mean. Ideally, a model respecting the given length constraint for each tested sample should have the entire distribution below the blue line.

As depicted in Figure 6, CoT based LLMs tend to produce long answers if not explicitly constrained, significantly impacting the generation time. The imposed length constraint in the CCoT prompt significantly affects the output length, although in practice LLMs are not always able to respect the given limit, especially for smaller values, such as 15, 30, or 40, which are more challenging.

To summarize, given the nature of the CCoT prompting, it is reasonable to consider a tolerance margin in respecting the requested length. To this end, in the following paragraphs we evaluate the considered models by the metrics proposed in Section 4, which extend the accuracy by also accounting for conciseness.

8 Final Remark and Conclusion

Limitations and Future Directions. From the findings revealed by the conducted experiments, a key insight is that for sufficient-scale models, such as Falcon-40b and Llama2-70b, CCoT effectively achieves a better trade-off between accuracy and efficiency. However, we acknowledge that smaller models may struggle to improve this trade-off, often producing incorrect answers when attempting to limit reasoning length. A detailed analysis of these findings is provided in the appendix B. We believe this aligns with the inherent capability of LLMs to exhibit a sense of understanding of output length in their generated responses(Bhargava

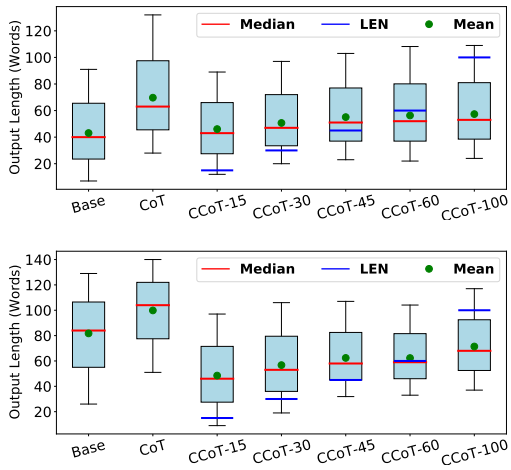


Figure 6: Distribution of output lengths (y-axis) between the 5th and 95th percentiles for different models and prompting strategies using the GSM8K test set. The top plot shows the results for Falcon-40b, while the bottom plot presents those for Llama2-70b.

et al., 2023). Future investigations should explore addressing this constraint and proposed metrics not only through inference prompts but also as part of a fine-tuning strategy, which could prove beneficial even for smaller-sized models.

Another point involves a deeper analysis of other potential benefits of conciseness in LLMs, beyond a study of the efficiency, which is the main scope of this work. For instance, while most analyses justify conciseness through a reduction in the number of steps (see Section 7.4), a detailed examination of redundancy and information reveals that this behavior also emerges in scenarios where CCoT and CoT have a similar number of steps. We believe that these metrics can be further leveraged to explore additional benefits of concise generation, such as mitigating hallucinations or reducing error propagation. For instance, a lower Information Flow score suggests that the model retains only the logically necessary information required to reach a correct answer, step by step, while excluding superfluous details. This approach could help reduce the risk of error propagation by filtering out unnecessary or irrelevant information at each step (Li et al., 2024).

Conclusion. This work explored the importance of conciseness in answers generated by LLMs for text-to-text tasks, introducing three new metrics to evaluate both conciseness and correctness based on user-defined parameters. Additionally, it proposed a prompt engineering approach, Constrained Chain-of-Thought, to get a better control of output length of generated answer ,thus impacting the inference time. Furthermore, it deepened the analysis of conciseness by evaluating its significance within the generated reasoning steps, particularly in terms of redundancy and information flow.

Experimental results demonstrated how the proposed Constrained CoT (CCoT) effectively highlights the trade-off (RQ1) and presents a unified perspective through new metrics (RQ2), particularly when applied to large LLMs such as Falcon-40b and Llama2-70b. Furthermore, we provide an in-depth analysis of the concept of conciseness, the capability to control output length, and the importance of information within the generated reasoning steps (RQ3).

In conclusion, this work emphasizes the need to focus on the conciseness of LLMs by proposing novel performance metrics that evaluate both the correctness of the output and its length. Additionally, the proposed CCoT prompting offers a simple yet effective strategy to enhance conciseness, potentially paving the way for new research directions to make LLMs more predictable and efficient.