illustrate the benefits of the proposed metrics and scores in highlighting the trade-off between accuracy and computational cost. Specifically, the following research questions are investigated in the next experiments: $\boxed{\text{RQ1}}$ Is the CCoT approach beneficial in terms of efficiency and accuracy compared to classic CoT?;$\boxed{\text{RQ2}}$ Are the proposed metrics representative of showing this trade-off?; and $\boxed{\text{RQ3}}$ Are the CCoT answers actually more concise, and to what extent are LLMs capable of controlling the output length based on an explicit prompt request?

7.1 Experimental setup

All the experiments have been carried out with the Text Generation Inference (TGI) platform[3] on 8 NVIDIA A100 GPUs. Specifically, we evaluated large and open source pre-trained LLMs from Hugging Face[4], such as instruction-tuned models Falcon-40b-instruct and model trained, reinforced by utilizing private data, namely Llama2-70b-chat-hf (Touvron et al., 2023). Additionally, further experiments on smaller models were conducted to evaluate their capability in handling CCoT, with detailed results provided in Appendix Section B.

The experiments utilized three arithmetic reasoning datasets: GSM8k (Cobbe et al., 2021), SVAMP (Patel et al., 2021), and ASDIV (Miao et al., 2021), commonly used to assess models' mathematical inference and computational reasoning abilities. The GSM8k test set includes over 1.3k problems out of 8,000. The SVAMP test set contains 300 examples, and the ASDIV test set comprises 1.22k examples.

For all experiments, the effectiveness of CCoT was compared by assessing the selected LLMs both with and without CoT (base mode).

7.2 Impact of CCoT on Accuracy and Efficiency

This experiment was carried out to evaluate the impact of *CCoT* on computation time and accuracy ($\boxed{\text{RQ1}}$). In particular, the selected LLMs were evaluated on three datasets using plain prompt (base), *CoT*, and *CCoT* with different length constraints:15, 30, 45, 60, 100. The results are presented in Figure 3 for Llama2-70b (top) and for Falcon-40b (bottom), showing accuracy and generation time in the first and second rows, respectively.

Considering the results with Llama2-70b, CCoT prompting demonstrates the ability to reduce generation time compared to CoT and, in most cases, achieves a time reduction similar to or better than plain prompting (base). Additionally, it generally improves or minimally impacts accuracy, thereby enhancing the trade-off in both directions. For instance, the average generation time decreases from 30.09 seconds with CoT to a maximum of 23.86 seconds with CCoT on GSM8K with Llama2-70b, achieved with a length constraint of 100, and further reduces with stricter constraints. At the same time, also the accuracy consistently improves, for example, with the GSM8k dataset, the accuracy of Llama2-70b increases from 36% with CoT to 37% (with CCoT-30) and 41.77% (with CCoT-100).

Similar observations can be made for the results with Falcon-40b (bottom part), where the CCoT approach improves the trade-off between efficiency and accuracy across all three datasets. In particular, CCoT achieves better accuracy overall for SVAMP and ASDIV, while significantly reducing generation time. This improvement in terms of accuracy, however, does not occur for the GSM8K dataset, where the sentences are more complex than those in the other two datasets. We believe that such a complexity makes it more challenging for a medium-sized model like Falcon-40b to effectively handle the CCoT constraint. Nonetheless, the accuracy remains higher than the base mode, indicating that the CoT-based approach still provides benefits.

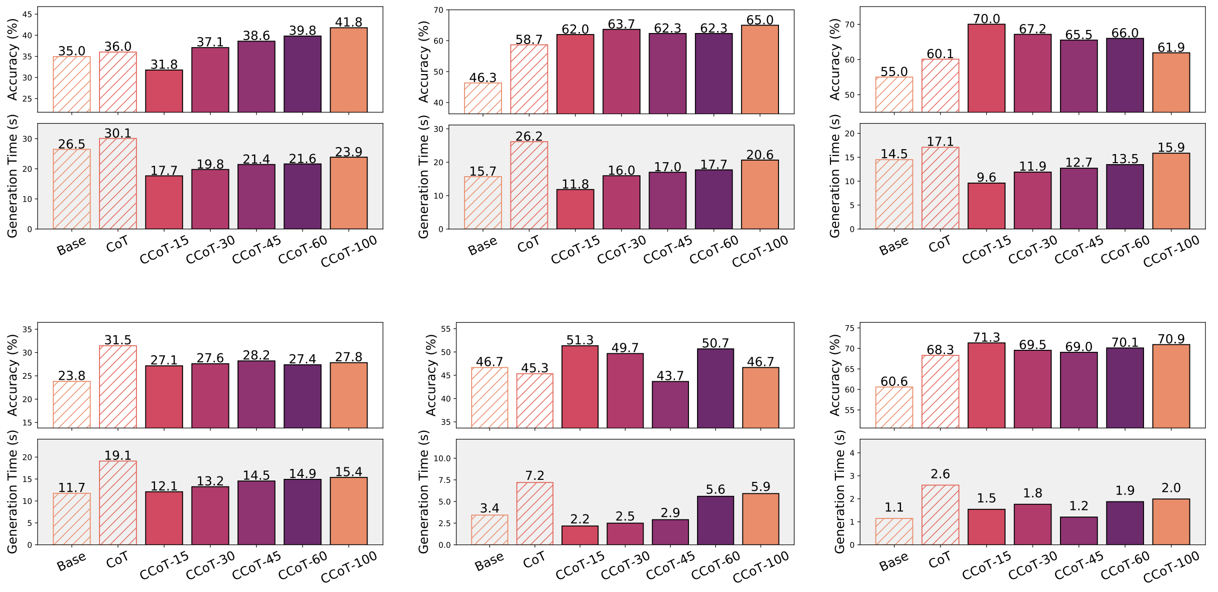
We also acknowledge that different behaviors may arise when dealing with datasets of varying complexity, as the nature of the questions can differ. For instance, CCoT-15 outperforms other CCoT variations because its 15-word responses potentially align better with the simpler nature of this dataset. We argue that this effect is closely related to the complexity of the questions, which makes the constraint impactful even on accuracy.

7.3 Analysis of the *correct conciseness*

Based on the previous results, CCoT demonstrates clear benefits in balancing accuracy and computational efficiency. To unify these aspects, we introduced new metrics in Section 4 that integrate answer length with correctness. The analysis presented in Tables 1 and 2 for Llama2-70b and Falcon-40b, respectively, highlights how these metrics effectively combine cost and accuracy ($\boxed{\text{RQ2}}$), while also remarking the advantages of CCoT.

[3] https://huggingface.co/docs/text-generation-inference

[4] https://huggingface.co/blog/os-llms



(a) GSM8k (b) SVAMP (c) ASDIV

Figure 3: Accuracy (white plots, the higher the better) and mean generation time (background colored plots, the lower the better) for Llama2-70b (top row) and Falcon-40b (bottom row) on the GSM8K, SVAMP, and ASDIV datasets. The models are evaluated using plain prompts (base), CoT, and CCoT under different len-constraints.

***HCA* evaluation.** The *Hard-k concise accuracy* evaluates the accuracy considering only the correct answers whose length is less than a specified value k. The top parts of Tables 1 and 2 report the value of this performance index across the three datasets, when using the different prompt approaches and for different values of k.

Specifically, for both Llama2-70b and Falcon-40b, the use of CCoT gets better results compared to base and CoT prompts across all values of k. Notably, for lower values of k, CoT prompts exhibit a significant reduction in performance, while this accuracy drop can be mitigated by using CCoT with strict length constraints, such as 15 or 30.

***SCA* evaluation.** We also evaluated both models using the *Soft Conciseness Accuracy (SCA)*, across different k values and α, where α represents a tolerance for accepting answers longer than the desired limit k. This metric is a generalization of the *HCA*, giving more flexibility in considering correct answers that are larger but still close to the desired length k.

The SCA values computed for Llama2-70b and Falcon-40b on the datasets are reported in center parts of the tables for different values of k and a fixed tolerance value $\alpha = 10$. For both models, the SCA values in CCoT settings are often comparable to HCA values for high values of k, such as 80 or 100. This is because, as discussed in Sec.7.5, for such lengths, the CCoT prompts are effective at returning outputs below the desired limit, making the tolerance less necessary. Conversely, for smaller k values, such as $k = 40$, SCA starts exceeding HCA, indicating that some correct answers have a length larger than k. However, for such k values of, using a tolerance α results in more pronounced improvements for CCoT prompts compared to Base and CoT. This means that, although many correct outputs are longer than k, under CCoT the model is still encouraged to constrain them close to k, thus achieving a higher score. This effect is particularly noticeable on Llama2-70b, which is more capable of controlling the length and produce correct outputs than Falcon-40b.

***CCA* evaluation.** The *Consistent Concise Accuracy* measures the capability of a model to generate correct answers whose lengths do not vary significantly, and therefore are consistent with the specified constraint. The *CCA* requires a third parameter β (in addition to k and α), denoting a tolerance on the output length variability. The bottom parts of the tables report the CCA scores obtained on Llama2-70b and Falcon-40b for $\alpha = 10$, $\beta = 40$, and different values of k, for the various prompting methods. According to these settings, the CCoTs results in a clear improvement

	Base	CoT	CCoT15	CCoT30	CCoT45	CCoT60	CCoT100
GSM8K - HCA							
H-∞	35.0	36.0	31.8	37.1	38.6	39.8	**41.8**
H-100	29.9	22.9	31.2	35.3	37.5	38.7	**38.9**
H-80	22.0	15.4	29.2	31.8	33.1	**35.0**	31.6
H-40	4.8	0.8	**12.7**	10.8	8.0	8.5	4.5
SVAMP - HCA							
H-∞	46.3	58.7	62.0	63.7	62.3	62.3	**65.0**
H-100	46.3	50.0	59.7	61.0	61.0	59.7	**61.7**
H-80	46.3	41.0	56.7	**57.7**	57.0	54.3	54.7
H-40	23.3	12.0	**44.7**	34.0	30.0	30.0	20.0
ASDIV - HCA							
H-∞	52.8	60.3	**67.2**	65.6	64.3	65.0	61.4
H-100	52.8	60.3	**67.2**	65.6	64.3	65.0	61.4
H-80	52.8	60.3	**67.2**	65.6	64.3	65.0	61.4
H-40	31.7	23.6	**55.5**	46.4	44.4	43.8	29.5
GSM8K - SCA ($\alpha = 10$)							
SCA-100	31.4	26.9	31.5	36.3	38.1	39.2	**40.1**
SCA-80	25.6	19.0	30.1	33.8	35.3	**36.6**	35.0
SCA-40	8.5	3.3	**18.0**	16.9	15.4	16.3	11.0
SVAMP - SCA ($\alpha = 10$)							
SCA-100	46.3	52.3	60.3	61.6	61.3	60.6	**62.7**
SCA-80	46.3	45.6	58.4	**59.2**	59.1	57.6	58.1
SCA-40	31.4	19.5	**48.2**	41.1	39.6	38.7	31.6
ASDIV - SCA ($\alpha = 10$)							
SCA-100	52.8	60.3	**67.2**	65.6	64.3	65.0	61.4
SCA-80	52.8	60.3	**67.2**	65.6	64.3	65.0	61.4
SCA-40	39.5	35.2	**60.4**	54.0	52.1	51.7	40.9
GSM8K - CCA ($\alpha = 10, \beta = 40$)							
CCA-100	31.4	26.9	31.5	36.3	38.1	39.2	**40.1**
CCA-80	25.6	19.0	30.1	33.8	35.3	**36.6**	35.0
CCA-40	8.5	3.3	**18.0**	16.9	15.4	16.3	11.0
SVAMP - CCA ($\alpha = 10, \beta = 40$)							
CCA-100	46.3	52.3	60.3	61.6	61.3	60.6	**62.7**
CCA-80	46.3	45.6	58.4	**59.2**	59.1	57.6	58.1
CCA-40	31.4	19.5	**48.2**	41.1	39.6	38.7	31.6
ASDIV - CCA ($\alpha = 10, \beta = 40$)							
CCA-100	52.8	60.3	**67.2**	65.6	64.3	65.0	61.4
CCA-80	52.8	60.3	**67.2**	65.6	64.3	65.0	61.4
CCA-40	39.5	35.2	**60.4**	54.0	52.1	51.7	40.9

Table 1: Llama2-70b

	Base	CoT	CCoT15	CCoT30	CCoT45	CCoT60	CCoT100
GSM8K - HCA							
H-∞	23.8	31.5	27.1	27.6	**28.2**	27.4	27.8
H-100	23.7	**29.3**	26.7	27.3	27.2	26.8	27.4
H-80	22.8	**26.8**	25.8	26.1	25.9	25.2	26.0
H-40	13.0	10.4	**13.4**	**13.4**	12.2	11.8	12.5
SVAMP - HCA							
H-∞	46.7	45.3	**51.3**	49.7	43.7	50.7	46.7
H-100	46.7	42.3	**51.3**	49.3	43.0	48.3	45.3
H-80	46.3	39.7	**49.7**	47.3	40.7	46.7	44.3
H-40	**38.7**	20.3	36.7	36.0	31.7	33.0	32.7
ASDIV - HCA							
H-∞	60.6	68.3	**71.3**	69.5	69.0	70.1	70.9
H-100	60.6	67.1	**71.2**	69.1	68.8	69.7	70.1
H-80	60.1	65.2	**70.9**	68.5	68.1	68.6	69.0
H-40	57.2	43.3	**64.9**	59.6	57.3	58.3	56.7
GSM8K - SCA ($\alpha = 10$)							
SCA-100	23.7	**30.1**	26.8	27.4	27.5	27.0	27.5
SCA-80	23.3	**28.1**	26.2	26.7	26.6	26.0	26.6
SCA-40	16.6	15.3	**18.0**	17.9	16.8	16.5	16.7
SVAMP - SCA ($\alpha = 10$)							
SCA-100	46.7	43.3	**51.3**	49.4	43.1	48.9	45.4
SCA-80	46.6	41.2	**50.7**	48.1	41.6	47.7	44.9
SCA-40	40.6	27.8	**42.7**	40.8	34.8	38.0	35.8
ASDIV - SCA ($\alpha = 10$)							
SCA-100	**84.4**	67.3	71.2	69.3	68.9	69.9	70.4
SCA-80)	**81.1**	66.1	71.0	68.8	68.5	69.2	69.5
SCA-40	66.0	50.4	**67.3**	63.3	61.5	61.9	61.2
GSM8K - CCA ($\alpha = 10, \beta = 40$)							
CCA-100	23.7	**30.1**	26.8	27.4	27.5	27.0	27.5
CCA-80	23.3	**28.1**	26.2	26.7	26.6	26.0	26.6
CCA-40	16.6	15.3	**18.0**	17.9	16.8	16.5	16.7
SVAMP - CCA ($\alpha = 10, \beta = 40$)							
CCA-100	46.7	43.3	**51.3**	49.4	43.1	48.9	45.4
CCA-80	46.6	41.2	**50.7**	48.1	41.6	47.7	44.9
CCA-40	40.6	27.8	**42.7**	40.8	34.8	38.0	35.8
ASDIV - CCA ($\alpha = 10, \beta = 40$)							
CCA-100	**84.4**	67.3	71.2	69.3	68.9	69.9	70.4
CCA-80	**81.1**	66.1	71.0	68.8	68.5	69.2	69.5
CCA-40	**66.0**	50.4	67.3	63.3	61.5	61.9	61.2

Table 2: Falcon-40b

compared to CoT prompting and base, for both Llama2-70b and Falcon-40b. However, for high CCoT length constraints (e.g., 100), the CCA score tends to decrease, which does not happen with the other two metrics. This can be explained by considering that an increased length constraint gives the model more freedom to generate outputs with higher variations, as discussed in Section 7.5.

7.4 Understanding the effect of CCoT

While the previous experiments demonstrate that CCoT achieves a clear reduction in the number of words produced, thus improving the trade-off between accuracy and generation time. The scores introduced in Section 6 are used in the following analysis to quantify the benefits in terms of conciseness (RQ3). As detailed in Section 6, the conciseness of an answer is assumed to depend on two key properties: *(i)* the number of steps required to produce a response, and *(ii)* assuming a similar number of steps, the amount of information repeated across those steps.

Number of Steps. To evaluate the first property, we plotted in Figure 4 the distribution of the percentage of generated answers based on the *number of reasoning steps* (Fu et al., 2022) for Llama2-70b (top row) and Falcon-40b (bottom row) across the three datasets. The results reveal a significantly different distribution in the number of steps across all datasets with Falcon-40b and for the ASDIV dataset with Llama2-70b. In fact, in these cases, the CCoT curves are clearly concentrated toward a reduced number of steps compared to CoT, indicating that the CCoT prompt consistently generates responses with fewer steps.

While the analysis of the number of steps explains the improved conciseness achieved with CCoT, more detailed analysis of the redudancy and information flow are required to understand the enhanced conciseness observed with Llama2-70b on GSM8K and SVAMP, where the number of steps follows a trend similar to that of CoT.

Redundancy Evaluation. As discussed in the previous paragraph, the distribution of reasoning steps for Llama2-70b on the GSM8K and SVAMP