Time is Fleeting - Applying Inference Constraints to Compact Models Transferred From Mathematical Reasoning to QA

Neil Chen
New York University
hc4549@nyu.edu

Nikolas Prasinos
New York University
np3106@nyu.edu

Sunny Son
New York University
sons01@nyu.edu

Terry Wu
New York University
tw3022@nyu.edu

Abstract

Large Language Models achieve strong results on question-answering (QA) tasks, but performance typically scales with model size. To enhance smaller models, we evaluate two techniques: budget forcing, which encourages extended reasoning at inference time, and fine-tuning on the s1k dataset, originally designed for mathematical reasoning. We apply these methods to the 8B-parameter LLaMA-3.1 model and assess their impact across three QA tasks: mCSQA (commonsense), Knights & Knaves (logical deduction), and REPLIQA (context retrieval). Our results show that dataset-specific fine-tuning consistently provides the largest gains. On K&K, both s1k fine-tuning and budget forcing significantly outperform the base model, demonstrating that mathematical reasoning skills can transfer to deductive QA. In contrast, REPLIQA performance degrades under budget forcing and sees limited benefit from s1k, likely due to its emphasis on fast, factual retrieval. For mCSQA, both approaches perform slightly better than the base model without though reaching the performance of fine-tuning, highlighting the limits of general reasoning augmentation in commonsense QA. These findings highlight the potential and limits of reasoning-based strategies for improving compact models on various QA tasks.

1 Introduction

1.1 Motivation

Large Language Models have revolutionized natural language processing through their remarkable ability to generate coherent and contextually relevant responses, especially in question-answering scenarios. However, the reliance on extremely large model architectures typically means increased computational resources and energy consumption, presenting substantial practical challenges for widespread deployment. Therefore, enhancing the performance of smaller-scale LLMs without significantly scaling model size has become a critical area of research.

A recent study (Muennighoff et al., 2025) proposed two complementary strategies for enhancing the reasoning abilities of smaller-scale language models. The first is a test-time method called budget forcing, which encourages models to engage in more deliberate and extended reasoning before producing an answer. The second involves targeted dataset curation, resulting in the creation of the s1k dataset, which is a carefully constructed benchmark of only a thousand data points, designed to foster specific mathematical reasoning skills. Together, these approaches enabled a Qwen-2.5-32B model to perform competitively on mathematical reasoning tasks, achieving results comparable to significantly larger models such as OpenAI's o1 and DeepSeek's r1 (DeepSeek-AI et al., 2025).

In this work, we investigate the applicability and effectiveness of these two strategies, budget forcing and s1k dataset fine-tuning, on a relatively smaller LLM, specifically the LLaMA-3.1-8B model (Grattafiori et al., 2024). We seek to determine whether these techniques, previously validated primarily on larger models, can similarly boost the performance of even smaller models across various challenging QA domains. Through systematic evaluations on commonsense, logical, and contextual QA tasks, we aim to provide insights into practical enhancements for small-scale LLM deployments.

1.2 Related Work

The ongoing need to replicate the performance of LLMs using smaller architectures has driven numerous studies and applications aimed at achieving comparable results with reduced computational requirements.

Recent advancements such as LLaMA (Touvron et al., 2023) have demonstrated that strategically training smaller-scale language models can yield

performance comparable to significantly larger models like GPT-3 (Brown et al., 2020), thus highlighting an essential avenue for making language models more accessible and efficient. LLaMA specifically focuses on optimizing model architecture and extensive training on publicly available data, showcasing strong capabilities even with relatively modest parameter counts. Another recent approach to enhancing small model performance (Hsieh et al., 2023), leverages chain-of-thought rationales from large models as additional supervision in training smaller models. Their results demonstrate impressive performance gains, with smaller fine-tuned models outperforming substantially larger models (e.g., a 770M T5 outperforming a 540B PaLM model), and achieving these results using significantly fewer training examples.

1.3 Budget Forcing and s1k Dataset

The work of Muennighoff et al. introduced a simple yet effective method called budget forcing, aimed at enhancing the reasoning capabilities of language models without altering their internal weights. The authors curated a specialized reasoning dataset called s1k, containing 1,000 carefully selected questions paired with detailed reasoning traces. This dataset emphasized diversity, difficulty, and quality to maximize reasoning improvement. After fine-tuning a Qwen-2.5-32B model on s1k, they implemented budget forcing at inference time by appending a special "Wait" token when the model attempted to prematurely conclude its reasoning, thereby prompting deeper thought, or truncating reasoning when it exceeded the necessary complexity. This allowed their model to dynamically allocate a reasoning budget, improving its accuracy through iterative self-correction.

Remarkably, this approach enabled the 32B model to surpass much larger models on challenging mathematical QA tasks. The integration of structured reasoning steps via Chain-of-Thought (CoT) prompts within the s1k dataset, which were distilled from Gemini 2.0 Flash Thinking Experimental, further facilitated deeper cognitive engagement during training, suggesting potential for broader applicability across various QA domains.

Inspired by this, our research explores the potential of applying budget forcing and fine-tuning on the s1k dataset to significantly smaller models, investigating whether these techniques maintain their efficacy and provide substantial reasoning enhancements to the LLaMA-8B model.

2 Methods

2.1 Experimental Setup

To assess the effectiveness of different strategies for enhancing small LLMs in QA tasks, we conduct controlled evaluations using several variants of the LLaMA-3.1-8B model across three question-answering domains: commonsense reasoning, logical deduction, and contextual understanding.

We begin with the base LLaMA model, evaluating its out-of-the-box performance without any fine-tuning or modifications. We then apply budget forcing at test time to this same base model to isolate the effects of inference-time reasoning guidance. In parallel, we fine-tune separate copies of the model on each target dataset—mCSQA, REPLIQA, and K&K—and evaluate them on their respective tasks to understand how direct task specialization influences performance. Finally, we explore whether reasoning abilities learned from the s1K dataset generalize across domains by evaluating a single model fine-tuned on s1K across all tasks, without further adaptation.

This experimental setup allows us to compare the benefits of inference-time reasoning control (via budget forcing) with task-specific fine-tuning and reasoning-transfer fine-tuning. The datasets used cover a diverse range of reasoning challenges, and their individual characteristics are discussed in the following section. The table below shows the outline of our experiments:

Model Configuration	mCSQA	REPLIQA	K&K
Baseline Evaluations			
LLaMA-3.1-8B (No Budget Forcing)	✓	✓	✓
LLaMA-3.1-8B + Budget Forcing	✓	✓	✓
Task-Specific Fine-Tuning			
Fine-Tuned on mCSQA	✓	-	-
Fine-Tuned on REPLIQA	-	✓	-
Fine-Tuned on K&K	-	-	✓
Cross-Task Transfer (s1K)			
Fine-Tuned on s1K	✓	✓	✓

Table 1: Overview of evaluation plan. A ✓ denotes that the given model configuration is evaluated on the corresponding dataset.

2.2 Budget Forcing Implementation

Our implementation of budget forcing closely follows the original paper, with a few minor adjustments to better accommodate the smaller-scale

model we use and the specific characteristics of our datasets.

Budget forcing operates by explicitly dividing generation into two stages:

- Thinking phases, which may be extended if the model stops too early.
- A final answer phase, which is triggered once a limit is reached.

It simulates a budget of thinking tokens allocated for iterative reasoning. After each reasoning phase, the budget is reduced based on the number of tokens used. The model continues generating new reasoning phases as long as the remaining budget allows. These iterations are bounded by a pre-specified maximum number of steps, typically set according to the model's computational limits. Once the thinking phase concludes, either by natural conclusion or by reaching the stopping conditions, a new prompt is constructed by appending an explicit finalization phrase to the previous reasoning. A final generation step is performed with a short maximum token limit to produce the model's answer.

More specifically, the technique is governed by two main parameters:

- `max_thinking_tokens`: the total maximum number of tokens the model may use
- `num_ignore`: the number of times the model is allowed to restart reasoning by appending a trigger token ("`Wait`") when it stops too early.

Extending Thinking

We begin by prompting the model to solve the QA task while encouraging step-by-step reasoning, with the prompt tailored to the specific task. If the model's initial output appears too short or shallow, budget forcing appends a "`Wait`" token to the current output and re-issues the prompt. This forces the model to pick up from where it left off and elaborate further.

This process can be repeated up to `num_ignore` times. After each repetition, the full prompt (including prior reasoning and "`Wait`") is sent back into the model for continuation. During this loop, the total number of tokens consumed is monitored against `max_thinking_tokens`.

In our implementation, based on preliminary experiments and constraints related to model size and available computational resources, we set the reasoning budget (`max_thinking_tokens`) to 1024 tokens and allowed the model to retry reasoning (`num_ignore`) up to 3 times.

Early Stopping of Thinking

To prevent the model from generating excessive, repetitive, or off-topic content, budget forcing employs early stopping rules. Specifically, if the total number of tokens generated during the reasoning phases exceeds the predefined budget (`max_thinking_tokens`), the loop is immediately terminated to avoid unnecessary computation and drift from the original task.

The flowchart (Figure 1) details the algorithm behind budget forcing in our implementation.

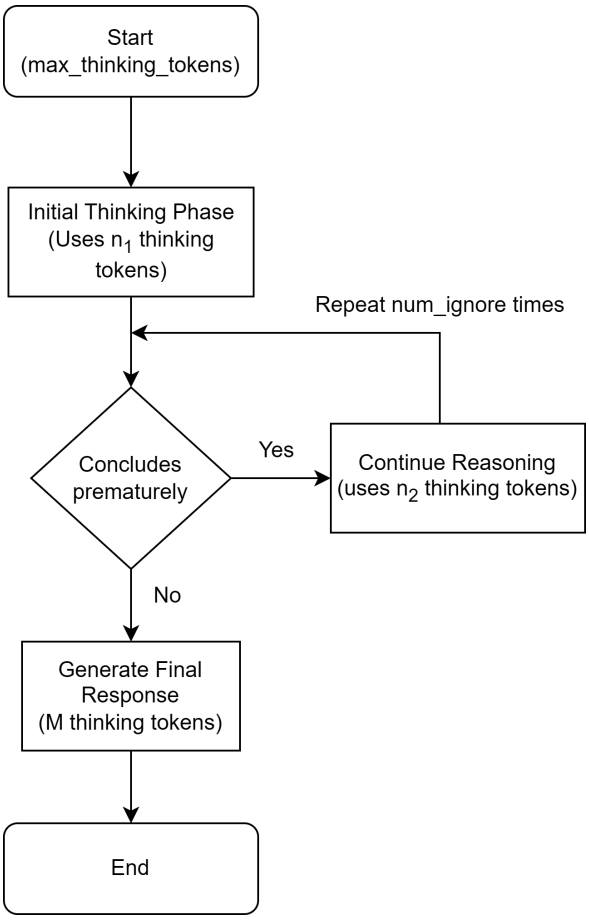


Figure 1: Budget Forcing Flowchart

2.3 Datasets

To evaluate the reasoning capabilities of our models across different QA tasks, we use three datasets that each emphasize a distinct aspect of reasoning (Table 2). Together, they provide a diverse and comprehensive benchmark for assessing the impact

of our proposed methods within the experimental setup.

Knights and Knaves (K&K)

The Knights and Knaves dataset (Xie et al., 2024) is a logical reasoning benchmark based on classic Knights and Knaves puzzles, where characters either always tell the truth (knights) or always lie (knaves). This dataset evaluates a model's ability to perform deductive reasoning in structured logical scenarios. Due to the inherent difficulty of this dataset we will only use the two-people puzzles.

Multilingual CommonsenseQA (mCSQA)

mCSQA (Sakai et al., 2024) is a cross-lingual commonsense dataset that tests contextual reasoning across multiple languages. Unlike translated datasets, mCSQA includes language-specific questions verified by human annotators. It uses multiple-choice questions requiring commonsense knowledge, focusing on linguistic nuances across eight languages. We will only be utilizing the English QA portion of this multilingual dataset.

REPLIQA

REPLIQA (Monteiro et al., 2024) is a context-based QA dataset featuring fictional documents to discourage reliance on memorized facts. Each document spans topics like cybersecurity, politics, folklore, or local news, paired with five questions. Twenty percent of queries are unanswerable, prompting models to recognize when no valid response exists. REPLIQA thus evaluates reading comprehension and context utilization.

s1K

The s1K dataset (Muennighoff et al., 2025) consists of 1,000 carefully curated questions designed to challenge a model's reasoning capabilities. Unlike conventional QA datasets, s1K is explicitly designed for reasoning-intensive tasks focused on mathematical problem-solving and logical inference. The dataset was created by selecting questions based on three key criteria: difficulty, diversity, and quality. This ensures that models trained on s1K develop strong general reasoning skills applicable across different domains.

2.4 Models

We adopt the LLaMA-3.1-8B-Instruct (Unsloth 4-bit) model as our base. This is a 4-bit quantized version of Meta's LLaMA-3.1-8B-Instruct model, optimized for low-memory environments and faster inference using the Unsloth (Daniel Han and team, 2023) framework. The instruct-tuned variant is aligned for conversational tasks and follows user prompts effectively out of the box. The 4-bit quantization allows us to efficiently conduct fine-tuning and inference on limited hardware without significant degradation in performance.

Dataset	QA Type	# Data
mCSQA	Commonsense	12273
REPLIQA	Context Retrieval	17955
Knights & Knaves	Logical deduction	300
s1K	Complex reasoning	1,000

Table 2: Number of available data points between the datasets. mCSQA, REPLIQA and Knights & Knaves are used for evaluation, while the s1k dataset is used for mathematical reasoning-based fine-tuning.

Fine-Tuning

We use the Unsloth framework to fine-tune our model. Unsloth is specifically designed for efficient training of large language models, supporting quantization-aware techniques like LoRA to reduce memory and compute requirements. We choose to fine-tune each model variant separately on individual QA datasets, rather than combining them, in order to preserve and amplify dataset-specific reasoning characteristics. Question-answering datasets often vary significantly in style, phrasing, and focus—training on them individually allows the model to better adapt to each dataset's unique structure without interference from competing task formats.

For the s1k dataset specifically, we append the reasoning trace generated by Gemini during the fine-tuning process. This decision was made to enhance the model's mathematical reasoning abilities, as one of our main goals is to compare its performance against the base model and the budget forcing implementation. In contrast, for the three QA datasets, no chain-of-thought traces are added during fine-tuning.

For each fine-tuning run, we use a single A100 GPU. We set both the per device train batch size and the gradient accumulation steps to 16 to manage memory usage while maintaining training stability. We train for 1 epoch on s1k, 1.5 epochs on mCSQA and REPLIQA, and 2.5 epochs on Knights & Knaves, as we observed that these datasets converge at different rates. It is worth mentioning that, for the K&K dataset specifically, we used data from

only two people (2ppl split), resulting in a relatively small training set of about 200 data points.

3 Results

We evaluate model performance across the three QA datasets. For each dataset, we assess the impact of four model configurations: the base LLaMA-3.1-8B model, the same model augmented with budget forcing at inference time, a variant fine-tuned on the s1k reasoning dataset, and models fine-tuned directly on each respective QA dataset. The following subsections present detailed results and analysis for each dataset individually.

3.1 mCSQA

On the mCSQA dataset (Figure 2), we observe a clear performance gain when applying fine-tuning directly on the task-specific data, with accuracy reaching close to 88%, significantly outperforming the base model at around 72%. Budget Forcing yields only a marginal improvement over the base model, indicating limited benefit from forcing extended reasoning on this dataset. In contrast, fine-tuning on the s1k dataset, which targets mathematical reasoning, achieves moderate transfer gains, improving accuracy to approximately 77%, but still not reaching the level of improvement that fine-tuning does.

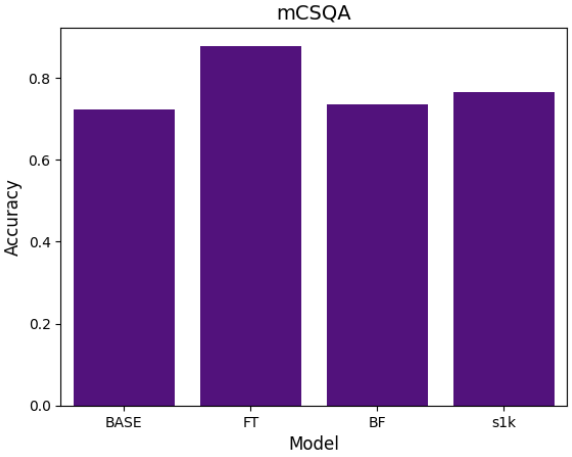


Figure 2: Model accuracy comparison across mCSQA

These results suggest that while direct fine-tuning is most effective for mCSQA, s1k transfer provides reasonable gains, and budget forcing offers minimal advantage in this context.

3.2 K&K

On the Knights & Knaves dataset (Figure 3), the base model performs just above chance level, achieving 28% accuracy where random guessing would yield 25% given four possible label combinations in the 2-person setup. Fine-tuning directly on the dataset substantially boosts accuracy to over 41%, while fine-tuning on the s1k dataset achieves a nearly identical improvement, indicating strong transferability of mathematical reasoning to logical inference. Budget Forcing improves slightly over the base model, reaching 35%, but still lags behind the targeted fine-tuning approaches, suggesting that enforced extended reasoning alone may not suffice for this type of deductive task.

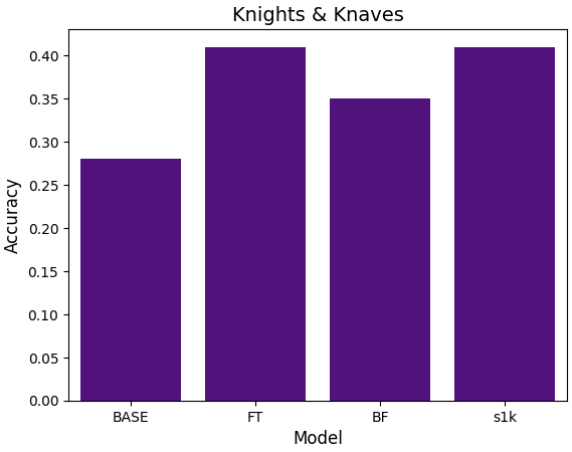
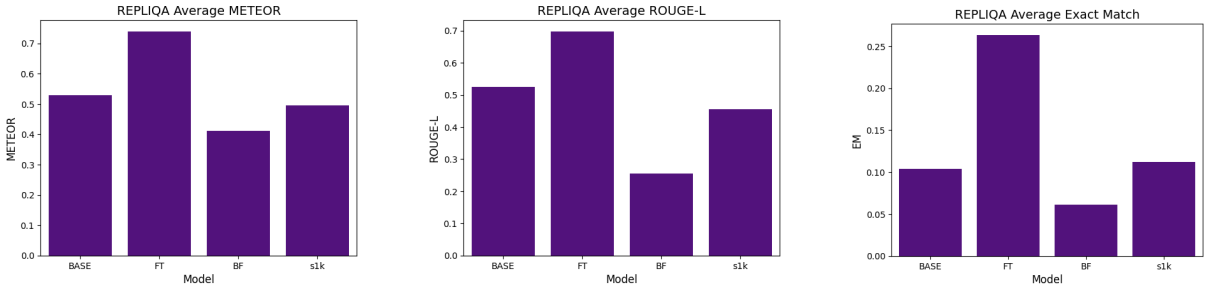


Figure 3: Model accuracy comparison across K&K

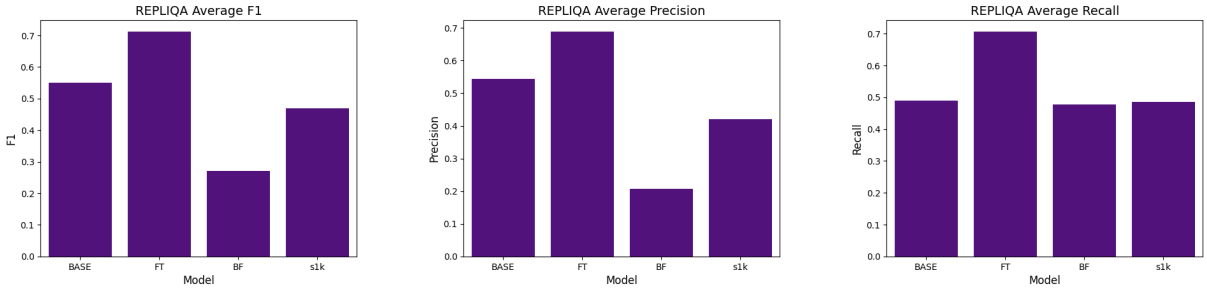
3.3 REPLIQA

On the REPLIQA dataset (Figure 4), we evaluate model performance using six standard text generation metrics: Exact Match (EM), F1, METEOR, ROUGE-L, Precision, and Recall. These are the same metrics, that the authors of the original dataset paper also used on their work. Exact Match captures the strictest form of correctness, requiring the predicted answer to match the reference exactly. F1, Precision, and Recall offer more lenient token-overlap-based measures, assessing how much of the predicted content aligns with the reference. METEOR and ROUGE-L further account for stem, synonym, and longest common subsequence overlap, making them well-suited for evaluating natural language generation tasks.

Across all metrics, fine-tuning leads to substantial improvements over the base model, with EM increasing from 10% to 26%, and F1 improving



(a) METEOR (b) ROUGE-L (c) Exact Match



(d) F1 (e) Precision (f) Recall

Figure 4: Comparison of REPLIQA metrics across different models.

from 55% to over 70%. METEOR, ROUGE-L, Precision, and Recall similarly increase by 15–20 percentage points, reflecting enhanced fluency and alignment with gold answers. In contrast, Budget Forcing consistently underperforms, in some cases dropping below base (Precision and ROUGE-L), suggesting that forced reasoning can degrade response quality in generative tasks. Finally, s1k-based fine-tuning offers modest gains across metrics, indicating partial transferability from mathematical reasoning, but remains notably less effective than direct task-specific fine-tuning.

The results clearly indicate that techniques aimed at enhancing reasoning, do not translate well to this task. REPLIQA is fundamentally a context retrieval dataset, requiring quick and accurate extraction of factual information from provided passages rather than step-by-step reasoning. Models fine-tuned on s1k or forced to think longer through budget forcing underperform, suggesting that these reasoning-heavy approaches may introduce unnecessary complexity or distract from the core retrieval task, which benefits more from direct, task-aligned training.

4 Discussion

4.1 Findings

Our results confirm that dataset-specific fine-tuning remains the most effective strategy for enhancing QA performance on specialized tasks. Fine-tuning directly on each QA dataset significantly outperforms all other approaches, yielding substantial gains such as over 15% improvement in mCSQA accuracy and nearly tripling the Exact Match score in REPLIQA compared to the base model.

Fine-tuning on the s1k dataset also yields notable improvements over the base model, particularly on tasks involving structured reasoning. On mCSQA and K&K, s1k fine-tuning leads to gains of +4% and +13%, respectively. The improvement on K&K is especially meaningful given its focus on logical deduction, which is a domain closely aligned with the mathematical reasoning skills emphasized by s1k. These results suggest that reasoning skills acquired through mathematical tasks can transfer effectively to logically structured QA tasks.

Budget forcing also provides some benefit when used with tasks involving structured reasoning. On K&K, it improves performance by +7% over the base model, and on mCSQA, it yields a modest gain of +1.3%.

4.2 Limitations

Despite the promising results, there are notable limitations to the general applicability of reasoning-based augmentation strategies. Budget forcing, in particular, degrades performance on REPLIQA, where the F1 score drops by over 50% relative

to the base model. This performance collapse suggests that forcing the model to "think more" may misalign with tasks that require direct, factual retrieval rather than extended reasoning. We observe several instances where the model initially produces correct reasoning chains but then second-guesses itself, ultimately generating incorrect answers.

Furthermore, budget forcing introduces inherent variability across runs due to the model being prompted to elaborate without explicit control over the content or focus of its continued reasoning. This can lead to redundancy, contradictions, or drift from the intended task. These effects were especially pronounced in REPLIQA and to a lesser extent K&K, where the task format is particularly vulnerable to overgeneration.

4.3 Future Work

One promising direction is to explore the full pipeline proposed by Muennighoff et al. in the original s1 paper, which combines dataset augmentation with reasoning traces and budget forcing. Instead of applying budget forcing directly to a base model, this pipeline first enriches training data with structured reasoning (for example using Gemini Experimental Thinking API), then fine-tunes the model on these augmented datasets before applying budget constraints at inference. Adopting such an approach on our QA datasets, could mitigate the negative effects observed in our experiments and improve generalization to more complex or nuanced QA tasks.

Additionally, while the reported results are consistent and reflect the general performance trends across methods and datasets, we note that a formal error analysis was not conducted due to limitations in computational resources and time. It would be beneficial in the future to include a formal error analysis to better understand failure modes across tasks and methods.

Lastly, incorporating more targeted stopping criteria and lightweight supervision during reasoning could help stabilize inference-time strategies like budget forcing, especially when applied to compact models. One promising direction is to introduce an external LLM agent to evaluate intermediate reasoning steps, providing feedback or early stopping signals before final answer generation. Another avenue is to aggregate multiple reasoning traces into a single, concise summary before concluding, which may reduce redundancy and guide the model toward more coherent and accurate outputs. These approaches could offer more controlled and interpretable reasoning without incurring excessive computational overhead.

5 Conclusion

In this work, we explored two strategies for enhancing the performance of compact language models on diverse QA tasks: inference-time budget forcing and fine-tuning on the s1k mathematical reasoning dataset. Through a systematic evaluation on three distinct QA benchmarks, commonsense (mCSQA), logical deduction (K&K), and context retrieval (REPLIQA), we found that task-specific fine-tuning remains the most effective approach, consistently yielding the highest performance across all settings.

Nevertheless, our results show that reasoning-enhancing methods like s1k fine-tuning and budget forcing can offer meaningful improvements in tasks requiring structured inference. Notably, both techniques improved performance on the K&K dataset, demonstrating the transferability of mathematical reasoning to logical deduction. However, their benefits do not generalize uniformly: for mCSQA, gains were modest, and in REPLIQA, budget forcing degraded performance significantly. These findings suggest that reasoning-based augmentation methods must be carefully aligned with the underlying structure and demands of the target task. For context-heavy or retrieval-based QA, step-by-step reasoning may introduce noise or overgeneration, underscoring the importance of tailoring inference strategies to task characteristics.

Our work highlights both the potential and the limitations of applying general-purpose reasoning constraints to small models, and motivates future research into more adaptive and content-aware inference techniques.

6 Author Contribution Statement

The work on this project was distributed equally among the four team members, aligning with each individual's strengths and interests. Each member took ownership of their respective responsibilities while maintaining a collaborative approach to ensure the smooth progression of the project. Given the demands of the work, we divided the tasks into three main parts:

6.1 Model Fine-Tuning

Sunny Son and Neil Chen led the fine-tuning of LLaMA models on the QA datasets and the s1k dataset, by using the available resources NYU HPC Greene Cluster. This component required two people due to the time-intensive nature of model training, as well as the challenge of identifying optimal hyperparameters and training configurations.

6.2 Budget Forcing Implementation

All team members collaborated on the implementation of budget forcing. The s1 paper and the accompanying GitHub repository were thoroughly studied to develop our own implementation of the technique. The nuances of the original method were faithfully incorporated while being adapted to our specific experimental setup.

6.3 Model Evaluation

Nikolas Prasinos and Terry Wu were responsible for model evaluation. After fine-tuning was complete, they conducted inference and evaluation across the three QA datasets. They also created the result visualizations, ensuring clarity, consistency, and visual quality.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Michael Han Daniel Han and Unsloth team. 2023. Unsloth.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. 2024. The llama 3 herd of models.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes.

Joao Monteiro, Pierre-André Noël, Étienne Marcotte, Sai Rajeswar Mudumba, Valentina Zantedeschi, David Vazquez, Nicolas Chapados, Chris Pal, and Perouz Taslakian. 2024. Repliqa: A question-answering dataset for benchmarking llms on unseen reference content. In *Advances in Neural Information Processing Systems*, volume 37, pages 24242–24276. Curran Associates, Inc.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling.

Yusuke Sakai et al. 2024. mcsqa: Multilingual commonsense reasoning dataset with unified creation strategy by lms and humans. In *Findings of the Association for Computational Linguistics (ACL)*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Chulin Xie, Yangsibo Huang, Chiyuan Zhang, Da Yu, Xinyun Chen, Bill Yuchen Lin, Bo Li, Badih Ghazi, and Ravi Kumar. 2024. On memorization of large language models in logical reasoning.