PSEUDO CHAIN OF THOUGHT INDUCED FINE TUNING FOR LARGE LANGUAGE MODELS

Pranav Srinivasa (Corresponding Author)
Dept. Computer Science and Engineering
BMS Institute of Technology and Management
Bangalore, Karnataka - 560064

Ajith S
Assistant Professor
Dept. Computer Science and Engineering
BMS Institute of Technology and Management
Bangalore, Karnataka - 560064

Pratham P Punneshetty
Dept. Computer Science and Engineering
BMS Institute of Technology and Management
Bangalore, Karnataka - 560064

Pratham Gowda
Dept. Computer Science and Engineering
BMS Institute of Technology and Management
Bangalore, Karnataka - 560064

Niveditha Nayana P
Dept. Computer Science and Engineering
BMS Institute of Technology and Management
Bangalore, Karnataka - 560064

Abstract:

Keywords: Large Language Models, Fine-Tuning, Chain of Thought, Reasoning, Natural Language Processing, Deep Learning, Machine Learning, Artificial Intelligence

The current state of Large Language Models (LLM) is filled with hallucinations and miscalculations, since the probabilities of the initial prompt and generations are not completely aligned with the probability distributions of the hypothetical right answer. Therefore, a chain of thought process allows the model probability to align more closely to that of the expected output by reiterating over internal thoughts. The framework for inducing chain of thought to Large Language Model is to, first fine tune the main LLM on specific data required and further a smaller adapter model or also called as Tiny Thought Model is obtained from two-step model distillation process to get a model in the order of 100M-1B parameters that is focused on creating thoughts and questions based on inputs. Finally, the Main LLM and the Tiny Thought Model are joined together via a router that decides if a thought is necessary to reiterate over the model answers.

1. INTRODUCTION

Large Language Models fail to address and provide correct answers for tasks that are not as simple as token generations since the model probability distribution does not align with the probability distribution of the answer. This can be fixed by providing more internal questions to allow the attention mechanism to align the model outputs to more closely match the output for the expected answers. A solution for this can be achieved by chain of thought prompting that requires a lot of manual labor for each prompt and increases the token size of input, or by the usage of large models that have internally adopted CoT such as the o1 and Claude models which have large param count, resource intensive and cost ineffective. Finally, direct CoT finetuning of smaller LLMs serves as a solution for the problem but may cause catastrophically forgetting of the knowledge gained, hence the proposed framework adopts regular LLM fine tuning along with adapter LLM of the order of 100-1B parameters to be joined via a router to regulate the internal thoughts and serve as a safe induction of Pseudo Chain of Thought into smaller LLMs for specific data. This method has the advantages of leveraging Chain of Thought for better outputs, with no knowledge loss and a low inferencing costs due to its smaller parameter count.

1.1. MOTIVATION

Large Language Models need to be smaller and easier to inference while having the ability to model the probability distribution of complex queries for effective usage for user specific data. We are trying to Improve the user experience with LLMs for company\user specific data, allow lower inferencing costs of LLMs and the ability for hosting locally, Leverage advanced mechanisms such as chain of thought at low inferencing and resource costs, making Large Language Models accessible and small, mitigate hallucinations of small LLMs by adopting Chain of thought. During the process of constructing this, it is crucial to optimize the data and architecture and training of these LLMs so that they can effectively model complex probability distributions without the need of massive amounts of resources. This includes having a balance between reduction in model size and preservation of its capacity for understanding complex user queries. The project also seeks to ensure that these smaller LLMs remain aligned with the specific needs of users and organizations, providing accurate and context-aware outputs.

1.2. OBJECTIVES

- To define the scope of the thought-specific model, Identify the domain, specify the tasks, and determine the representative dataset for specialization.
- To distill the large model into a compact thought-specific model, use knowledge distillation techniques to transfer reasoning capabilities into a smaller model (100M–1B).
- To implement an adapter framework for specialization, incorporate lightweight adapter layers into the large model and adapt it for the specific domain.
- To generate synthetic question-answer pairs, Leverage the large model to create diverse, high-quality Q&A datasets from document-specific data.
- To fine-tune the large model on the generated synthetic data, Train the large model on Q&A datasets to enhance domain-specific reasoning and task performance.
- To design a router neural network for reasoning depth, create a model to determine the optimal number of intermediate thoughts required for specific tasks.
- To train the router to connect models dynamically, Train the router to decide whether to use the large model or the thought-specific model for each query.
- To integrate the router with the fine-tuned and distilled models, Ensure seamless connection between the large model, the router, and the thought-specific model.
- To test the complete system for accuracy and efficiency, Validate the integrated system's outputs and measure its performance across diverse tasks.
- To iteratively refine the system based on performance feedback, continuously improve the models, router, and workflows to meet evolving requirements and optimize task outcomes.

2. METHODOLOGY

This is novel approach to induce a reasoning or thought process to any pre-trained large language model with small task specific datasets while also maintaining the knowledge base integrity of the pre-trained model. The proposed methodology can quantitatively improve the accuracy within a large language model can respond with, since the probability density of the pre-trained model gets aligned to the hypothetical target probability through the reasoning steps involved.
In the proposed approach the first step involves generating a synthetic dataset that only involves the question and a reasoning output for a specific question. This dataset is employed into training an 8-13B (Billion) parameter Large Language Model. The training process involves the use of the Low Rank Adaptation [11] (LoRA) method. After the process of finetuning the Large Language Model to perform reasoning tasks, the knowledge is distilled using the Knowledge Distillation method employing KL-Divergence for Large Language Models [13]. This process involves a two-step distillation from the original 8B-13B model to a 3B model initially and then to a 1B model. This is done so that knowledge gained from the 8B-13B model can be easily passed down to a smaller model whereas a smaller model would be incapable of learning all the complex relations that can be modeled by the larger model. The resultant 1B LLM is modeled as the Tiny Thought Model or the TTM. Furthermore, another base-LLM is chosen to be the main responder of the model. This Base-LLM is finetuned using the LoRA approach for any task specific use cases such as math, computer science, coding etc. A Router which is a feed forward network is trained based on the responses of the Base-LLM to decide when the optimal answer is reached compared to the question. Finally, the Tiny Thought Model and the Base-LLM are attached via the Router and hence Chain of thought is induced into the base-LLM. During the process of a query asked by the user, the Tiny Thought Model first generates reasoning steps to be followed by the base model. This is then passed onto the base-llm and a response it given.

3. MODEL ARCHITECTURE

The model architecture consists of a Tiny Thought Model (TTM) with 1B-100M parameters, a fine-tuned LLM with 7-13B parameters and a Router which is a feedforward MLP network. They are connected such that the overall system mimics a

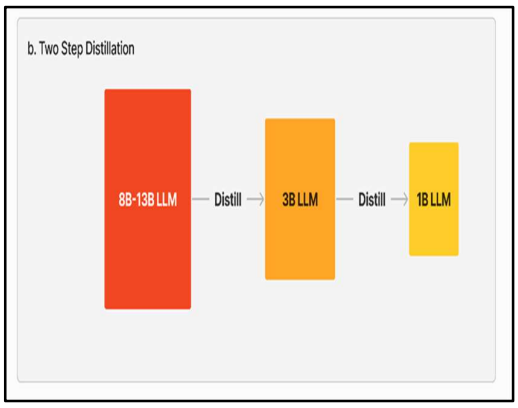

Fig. 3. 1 Two Step Distillation Process

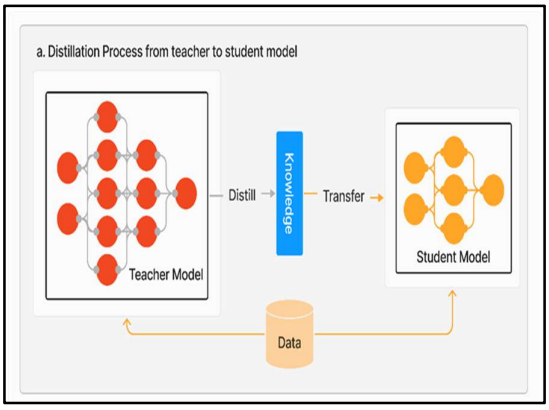

Fig. 3. 2 The Distillation Process

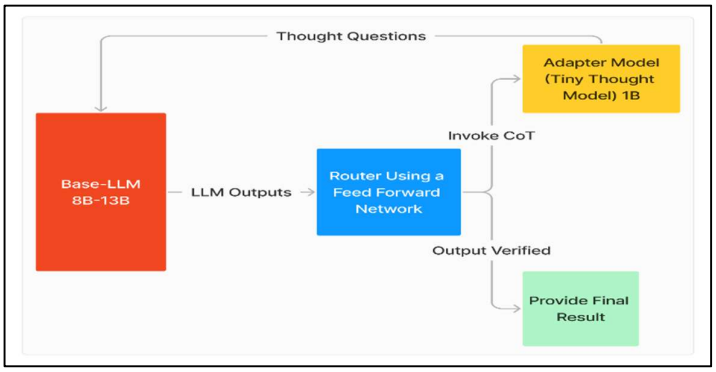

Fig. 3. 3 Overview of the Complete Model Architecture

Chain of thought structure with thoughts and reasoning cycles before providing the final answer, increasing the accuracy of alignment of model probabilities with real world probabilities. Fig. 3.1 represents the process of obtaining the Tiny Thought Model for a Large Language Model of the order 8B-13B parameters. The process involves Two step distillation to avoid any knowledge that may occur in direct distillation from 8B-13B param model to a 100M-1B Tiny Thought Model. In this process of Two Step distillation involves first distillation from the Large Language model to an intermediate student model that is of the order 7B-13B params, further the intermediate student model is distilled again to obtain the Tiny Thought Model of 100M-1B parameter that is capable of closely matching the probability distribution of the parent model. The knowledge distillation as shown in Fig. 3.2 loss function combines the supervised loss with the knowledge distillation loss in a weighted binary exponential format. Fig. 3.3 refers to the overall system architecture of the pseudo chain of thought fine-tuned model. Here, the base Large Language Model (LLM) is fine tuned to on the synthetic data from the document, further a trained Feed forward network is trained to invoke chain of thought optimal number of times before providing final improved response. This Feed Forward Network also called the Router is used to join the fine-tuned base LLM with the Tiny Thought Model obtained through two step distillation process referenced in Fig. 3.1 Therefore, after the system is constructed as shown in Fig. 3.3, based on an input from the user, the base LLM provides an output, based on which the router decides to invoke chain of thought, further the Tiny Thought Model creates an internal thought/question which is passed on to base LLM along with previous outputs and queries. This loop continues until the router network decides to exit reaching an optimal answer.

4. IMPLEMENTATION

4.1. DATASET FORMATION

For the purposes of model distillation and creation of a Tiny Thought Model, a large dataset with reasoning steps for logical questions is to be generated. Here the target text for the model should not be the final answer and instead the intermediate reasoning steps. Hence several small datasets for math, science and computer science were collected and formatted with LLM tags, here we use Llama 3 suite of models hence the tags are <start_header_id>, <end_header_id>, <eot_id>, <end_message_id> and many more which signify certain format context to the LLM. After formatting all the datasets with system instructions, user responses and assistant responses, they are all merged and shuffled such that each subset contains the same percentage of each dataset's row.

4.2. HYPER PARAMETRIC SETTINGS

4.2.1. LoRA Fine Tuning of Base LLM:

For faster training and resource efficiency a 4-bit quantized Llama 3.1 8B model is opted to be the base LLM. With the help of Unsloth fine tuning a QLoRA finetuning framework was developed for the Base LLM with mathematics expertise dataset. Here, the Query, Key, Value Matrices along with Up and Down projections were enabled for the fine-tuning process. It utilized a binary float 16 datatype for weights with a linear learning rate scheduler with the target learning rate being $2 * 10^{-4}$. The rank was set to 32 and alpha to 64 with the batch size being 8 per device. Due to resource constraints the maximum step size was set to 300.

4.2.2 Knowledge Distillation for TTM:

For the purposes of knowledge distillation, the process of normal finetuning took place for a Llama 3.1 8B with a reasoning specialized dataset and hence was used in finally distilling to a Llama 3.2 1B model. To implement Knowledge Distillation a standard Supervised Fine-Tuning Trainer was implemented with the compute class being overridden to implement the KL Divergence knowledge distillation loss. Here, only the Query and Key matrices were enabled for distillation and a batch size of 2 with a maximum training step size of 200 was implemented. Learning rate schedulers were similar to the parametric settings in LoRA finetuning process mentioned in section 5.2.1.

4.3. INFERENCE AND EVALUATION

For the purposes of evaluation, the Base LLM was set to a greedy decode method, with a top K being 100 and top P being 0.95, due to resource constraints. For inference purposes the Base LLM was set to a beam search decode method with the number of beams being 2. The Temperature for the base LLM was set to 0.6 with a max new token of 2048, during both evaluations and inference to promote more accurate results. The Tiny Thought Model was set to a beam search decode method with the number of beams being 2 and a max new token of 512 during both evaluations and inference. For evaluation and training both the models were prefixed with certain system instructions and a 1 shot example for the output format. All the above settings were constant for all the comparison models used to establish a control in the environment.

5. RESULTS AND DISCUSSION

S.NO.	MODELS	MMLU COLLEGE MATHS			MMLU COLLEGE CS		
		ACCURACY %	BLEU	ROUGE	ACCURACY %	BLEU	ROUGE
1	Llama 3.1 8B	22	0.1300	0.0500	49	0.1700	0.1900
2	Llama 3.1 8B CoT	32	**0.5300**	**0.5000**	**50**	**0.6800**	**0.6800**
3	Llama 3 8b math finetuned	26	0.1298	0.0313	22	0.0590	0.0201
4	Llama 3 8b math finetuned with CoT	**37**	0.1158	0.0137	33	0.0652	0.0233
5	Mistral 7b v0.3	27	0.1174	0.0139	30	0.0656	0.0203
6	Gemma 2 9B	29	0.1134	0.0129	39	0.0615	0.0190

Table 5. 1:Quantitative results of evaluation over selected benchmark

Finally, after the process of two step knowledge distillation using a combination of cross entropy loss and reverse kl divergence loss, we obtain a Tiny Thought Model of size 1B reduced from 8B. The Base LLM can be unaltered but for task specific applications such as math and reasoning fine tuning on an appropriately large dataset for the task improves the overall answering capacity and internal knowledge in the probability distribution of the LLM.

5.1. METRIC DESCRIPTION.

After Combining both these models as mentioned in the proposed methodology, evaluation on standard datasets such as Massive Multitask Language Understanding Meaning (MMLU), with the college math and college computer science split, is carried out with metrics being Accuracy, bilingual evaluation understudy (BLEU), Recall-Oriented Understudy for Gisting Evaluation (ROUGE). The Bilingual Evaluation Understudy (BLEU) has gained a lot of popularity as a metric used in evaluating machine translators as well as text generation models. It calculates n-gram matches with various reference texts and implements a precision measure at the same time taking into account a brevity penalty for short generated outputs.The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) on the other hand is used for the assessment of text summarization and generation on a recall basis with n-grams overlapping. Some of the families which ROUGE is composed of are ROUGE-N, ROUGE-L, and ROUGE-W which check for n, longest and weighted sequences respectively

5.2. COMPARISON OF MODELS

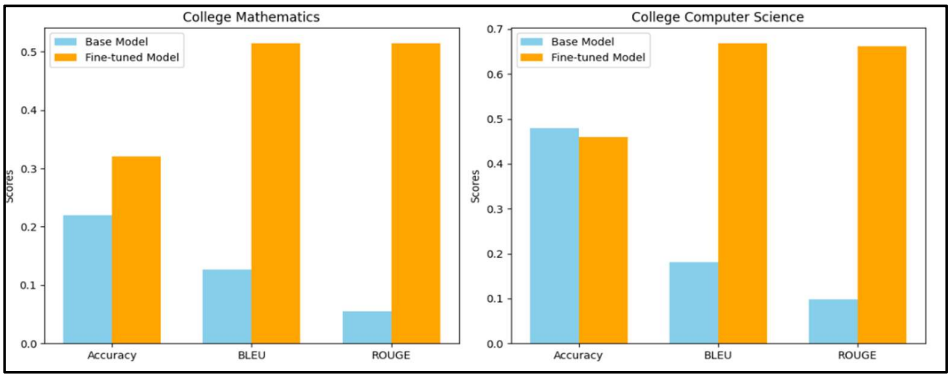

Fig. 5.1: Evaluation results to compare Proposed LLM vs BASE LLM

The Table 5.1 depicts the values achieved by each of the selected models for the benchmark of MMLU Mathematics and MMLU Computer Science, each being at college level. The Models chosen for the purpose of comparison involves the base models (Llama 3 8B) to establish a control and some commercially available models of comparable parameter counts such as the Mistral 7B v0.3, Gemma 2 9B. The metrics used here are Accuracy percentage, BLEU and ROUGE L1. The highlighted values at each of the metric section for each of the selected benchmark depicts the highest score achieved in the respective benchmark for the respective metric. As observed the highlighted values are skewed towards models with an ability to have a chain of thought. Hence a Rudimentary analysis results in association of better performance with Chain of Thought. Hence with the proposed methodology the models with induced chain of thought have achieved higher

scores as compared to the control base large language models while also achieving better performance as compared to other open-source models of comparable parameter count such as Mistral 7b and Gemma 2 9b.

.

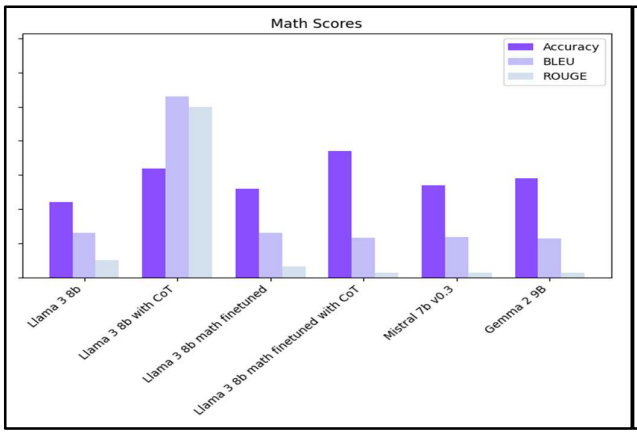

Fig. 5.2: Comparison of College Computer Science Scores with other models

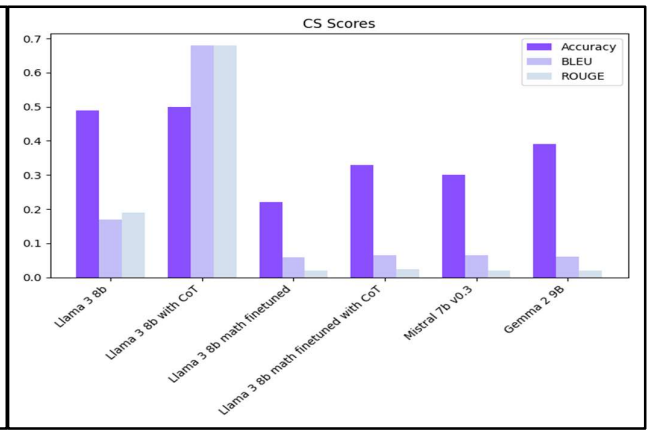

Fig. 5.3: Comparison of College Mathematics Scores with other models

The Fig. 5.1 shows the comparison between the base LLM and CoT fine-tuned LLM. This shows a significant improvement of the CoT induced LLM in math with a 10% increase in accuracy rates and significantly higher BLEU and ROUGE scores which indicate better explainability of LLM for the answer. This shows that even without any Knowledge finetuning of base LLM, and only by inducing Chain of Thought TTM, the model performance is perceivably better.The Fig. 5.2 Shows the comparison of the MMLU College level computer science evaluations of the proposed Pseudo CoT induced model with the publicly available models while using the base model as comparison. Here due to lack of fine-tuning knowledge of the distribution of data, the non-fine-tuned, Chain of Thought induced model performs the best while also performing better than all the other models such as Gemma 2 9B, Mistral 7B v0.3. The Fig. 5.3 also shows similar results but comparison of MMLU Mathematics yields the Fine-Tuned Pseudo CoT model as the victor due to training on similar distribution data. Hence this shows the models capability on task specific operations which require reasoning.

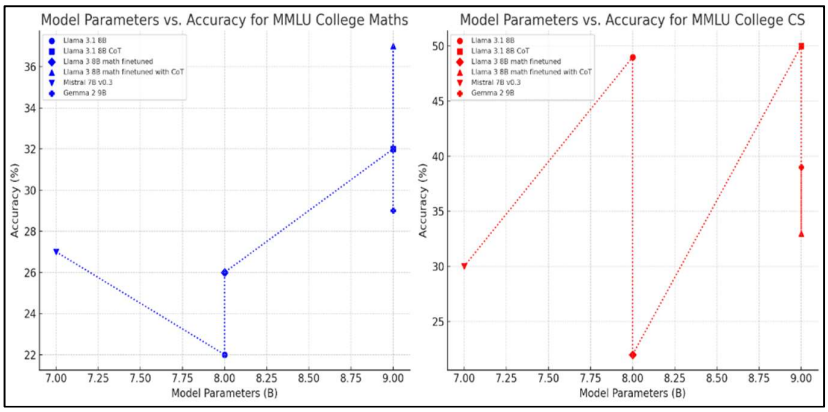

Fig. 5.4: Model Parameter vs Accuracy for each evaluation benchmark.

The Fig. 5.4. depicts the comparison between the model size over the model accuracy achieved on each of the bench marks. The models with CoT induced finetuning have been increased by 1B parameters accounting for the Tiny Thought Model. Even with this slight initial increase, we can deduce that there exists a large improvement over the accuracy compared to the base model aswell as the finetuned model, hence making the tradeoff between the between increased model size to accuracy is validated and justified.

Further compared to models of similar parameters size and higher parameter size compared to the base model chosen, it is observed to have a substantial improvement, hence validating the additional resources to induce chain of thought to a smaller model as opposed to using a commercially available larger model.

6. CONCLUSION

6.1. SUMMARY

The proposed Pseudo Chain of Thought Induced Finetuning of model is a significant advancement in improving the reasoning and task-specific capabilities of large language models (LLMs). Through the two-step knowledge distillation process, the Tiny Thought Model (TTM) is reduced from 8B to 1B parameters. Further, fine-tuning the Base LLM on task-specific datasets, particularly in mathematics and reasoning, has significantly improved the model's answering capability.Evaluation using the Massive Multitask Language Understanding (MMLU) dataset revealed key findings In mathematics performance a 10% increase in accuracy compared to the Base LLM is observed, with higher BLEU and ROUGE scores, highlights improved explainability and reasoning, Computer Science Performance without knowledge fine-tuning, the CoT-induced model outperforms all publicly available alternatives such as Gemma 2 9B and Mistral 7B v0.3.These results underscore the efficacy of combining the distilled TTM and a fine-tuned Base LLM for achieving superior performance in both general and task-

specific reasoning challenges but there exists a lot of scope for improvement of the current results which let the model outperform higher parameterized models as well.

6.2. FUTURE WORK

These improvements mainly are, scaling model size of the TTM and the Base LLM have substantial performance gains. Exploring TTM sizes of 3B or 4B could balance efficiency with reasoning depth, while scaling the Base LLM to 13B or larger. Increasing Training Time for the distillation and fine-tuning process, especially with larger datasets, can improve the alignment of the TTM's reasoning capabilities and the Base LLM's knowledge. Dynamic Routing Optimization between the TTM and the Base LLM to better identify tasks requiring detailed reasoning could speed up performance. Adaptive mechanisms could be developed for selecting reasoning depth based on task complexity. Specialized CoT Techniques exploring domain-specific Chain of Thought patterns.In conclusion the proposed model marks a substantial leap in reasoning and task-specific performance, paving the way for future advancements that scale both the architecture and its training paradigm.

6.4. REAL WORLD APPLICATIONS.

The combination of the two Large Language Models is negligible since the distilled Tiny Thought Model only constitutes to 1.2GB of additional memory in comparison to the ~ 6-7GB of memory utilized by the Main model.Hence due to the relatively small size and performance boost the combined model architecture can be used for various small to medium tasks, some of them being, Code base reviewing, Customer Care Agents, Customer Relations handling and many more.The small size and parallelizable properties of the proposed architecture allows the model to be self-hosted at minimal cost for small and medium tasks that are not at the highest of priorities and provide great cost-effective solutions compared to the utilizing Large Foundational Models of the order 50-100B for tasks that do not require large models and instead requires reasoning abilities. The integration of the Pseudo Chain of Thought (Pseudo-CoT) technique into Large Language Models (LLMs) offers several pragmatic real-world uses for sectors that require improved reasoning, efficiency, and cost savings. The combination of a compressed Tiny Thought Model (TTM) and Base LLM fine-tuned is intended to improve computational efficiency with high task completion accuracy rates.

REFERENCES

[1] Brown, T. B. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
[2] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., ... & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, *35*, 24824-24837.
[3] Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., ... & Le, Q. V. (2021). Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.
[4] Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., ... & Zhou, D. (2022). Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.
[5] Saparov, A., & He, H. (2022). Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *arXiv preprint arXiv:2210.01240*.
[6] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, *33*, 9459-9474.
[7] Perez, E., Lewis, P., Yih, W. T., Cho, K., & Kiela, D. (2020). Unsupervised question decomposition for question answering. *arXiv preprint arXiv:2002.09758*.
[8] Hoffman, M. D., Phan, D., Dohan, D., Douglas, S., Le, T. A., Parisi, A., ... & A Saurous, R. (2024). Training chain-of-thought via latent-variable inference. *Advances in Neural Information Processing Systems*, *36*.
[9] Kim, S., Joo, S. J., Kim, D., Jang, J., Ye, S., Shin, J., & Seo, M. (2023). The cot collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning. *arXiv preprint arXiv:2305.14045*.
[10] Mitra, C., Huang, B., Darrell, T., & Herzig, R. (2024). Compositional chain-of-thought prompting for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 14420-14431).
[11] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2021). Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
[12] Li, Y., Yu, Y., Liang, C., He, P., Karampatziakis, N., Chen, W., & Zhao, T. (2023). Loftq: Lora-fine-tuning-aware quantization for large language models. *arXiv preprint arXiv:2310.08659*.
[13] Gu, Y., Dong, L., Wei, F., & Huang, M. (2023). Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*.
[14] Lin, J., Tang, J., Tang, H., Yang, S., Chen, W. M., Wang, W. C., ... & Han, S. (2024). AWQ: Activation-aware Weight Quantization for On-Device LLM Compression and Acceleration. *Proceedings of Machine Learning and Systems*, *6*, 87-100.
[15] Zhang, Z., Zhang, A., Li, M., & Smola, A. (2022). Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.
[16] Lyu, Q., Havaldar, S., Stein, A., Zhang, L., Rao, D., Wong, E., ... & Callison-Burch, C. (2023). Faithful chain-of-thought reasoning. *arXiv preprint arXiv:2301.13379*.