Thinking Adapter
Enhancing LLM Reasoning with Low-Latency Thought Processing

Spencer Presley, Dustin O'Brien (Salisbury University)
Dr. Junyi Tu (Salisbury University)

Abstract

Large Language Models (LLMs) have been one of the biggest revolutions in artificial intelligence in recent history. There is numerous research published daily exploring new ways to improve them. A recent trend is adding a "reasoning" or "thinking" section prior to the actual response. This reasoning mode, seen in OpenAI"s O1, Deepseek"s R1, and models from Claude and Gemini, acts as a chain-of-thought layer that significantly improves LLM responses. We introduce a "thinking adapter" that can mimic this behavior and be attached to any model, including local ones. This approach enables smaller models like GPT-4o-mini or local models like Mistral NeMo to produce higher quality responses approaching those of larger models. Our thinking adapter is inspired by Tree of Thought and Graph of Thought as they show increases in problem solving ability, but is not built directly off of them. The reason we don't implement them directly is that while ToT and GoT showed significant improvement in LLM performance, they also resulted in high latency. To make our thinking adapter more usable for many tasks such as conversations, we instead use ToT's and GoT's ideas of exploring complex problem spaces in discrete steps, while introducing new implementation methods to optimize the adapter to achieve low latency while maintaining significant improvement in performance.

Objectives

- Improve local model performance via reasoning based on ToT
- Decrease ToT latency while maintaining performance
- Investigate a system which works better in more generalized problem spaces
- Provide a unified way to add this to any model.
- Investigate the performance of multi-modal models in a collaborative environment (i.e., models working together).

Methods

Our methodology followed a test-driven approach. We began by experimenting with Tree of Thoughts (ToT), which quickly revealed a significant drawback: it was extremely slow. In response, we tested smaller models to assess whether reduced model size would improve speed. However, we found that ToT remained slow regardless of the model used. This prompted us to theorize and prototype distilled alternatives. We developed two separate architectures:

1.Linear Reasoning Adapter: This was the simplest possible design. A user query passes through *n* sequential reasoning stages before entering the final model, whose output is returned to the user.

2.Distilled Tree of Thought (DToT): This more complex design instructs the LLM to decompose the input query into granular subtopics. These are structured into a tree, initially at a depth of 1. We then apply distillation by constraining the tree to a maximum width of 3 and a depth of 3—yielding at most 9 nodes. The AI traverses this tree, dynamically deciding which topics warrant further exploration.

Performance-wise, the linear adapter achieved a fast response time of 17.7 seconds per query. The DToT model, achieved an average response time of 184 seconds. For benchmarking, we also evaluated several baselines, including local LLaMA3.1 and Mistral Small models, as well as GPT-4o-mini.

We benchmarked the models using LlamaIndex, a framework for building and testing LLM applications.

Results

Figure 1: Linear Reasoning

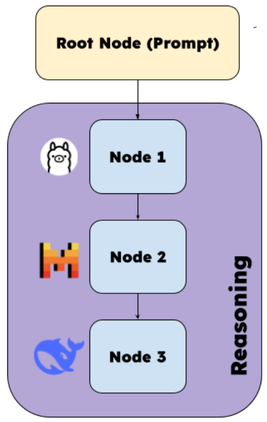


Figure 2: Distilled Tree of Thought

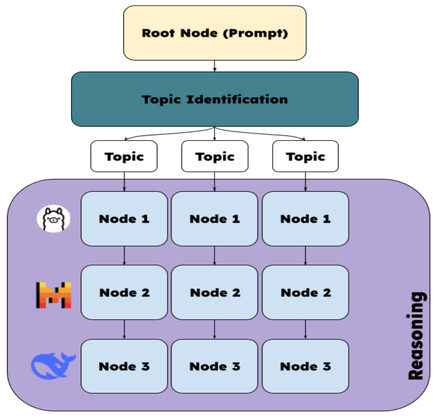


Figure 3
Model Correctness

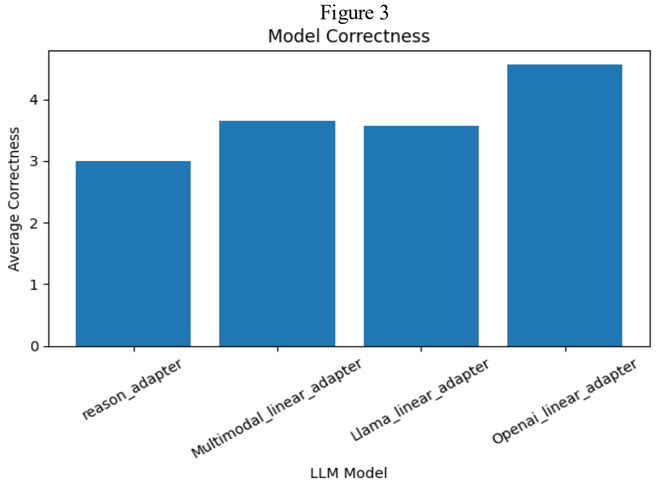
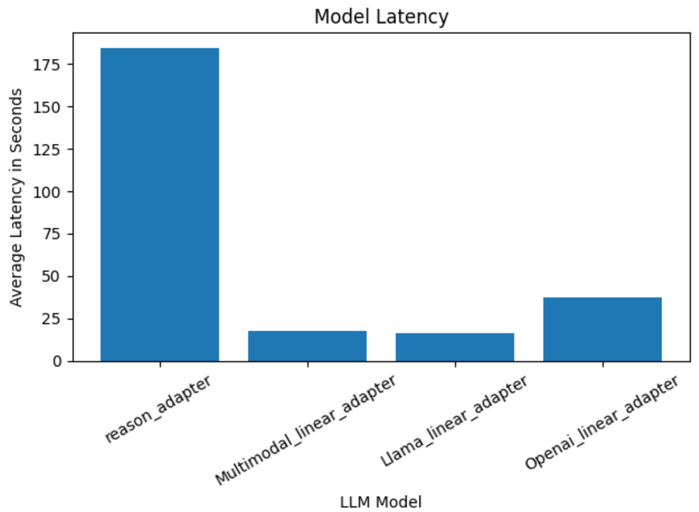


Figure 4
Model Latency



Conclusion

Our investigation into thinking adapters aimed to enhance LLM performance, particularly for smaller models, while mitigating the latency issues associated with complex reasoning structures like Tree of Thoughts. We developed and evaluated two primary architectures: a sequential Linear Reasoning Adapter (illustrated in Figure 1) and a more complex Distilled Tree of Thought (DToT) approach (illustrated in Figure 2), referred to as 'reason_adapter' in our benchmarks.

The results, visualized in Figure 3 and 4 demonstrate a clear trade-off. The DToT adapter ('reason_adapter'), designed for deeper reasoning via topic decomposition, exhibited significantly higher latency (~180 seconds) and achieved lower correctness scores compared to the linear variants in our benchmarks.

Conversely, the Linear Reasoning Adapters provided a more efficient solution. Implementations using Multimodal, Llama, and OpenAI base models achieved substantially lower latencies (~16s to ~37s). Notably, the 'Openai_linear_adapter' not only maintained low latency but also yielded the highest average correctness score (~4.3). This suggests that a simpler, sequential reasoning process can effectively improve response quality without prohibitive computational cost, particularly when paired with a capable base model.

In conclusion, while the DToT approach offers an interesting direction for complex problem decomposition, the Linear Reasoning Adapter presents a more practical method for adding beneficial reasoning capabilities to LLMs, effectively balancing performance gains with acceptable latency. This adapter framework successfully demonstrates a viable path towards enhancing smaller and local models, meeting our primary objectives of improved performance and usability. Future work could explore hybrid approaches or further optimize the DToT architecture to reduce its latency.

References

- Spencer Presley, Dustin OBrien. (2025). *COSC390_Benchmarking [Source code]. GitHub.* https://github.com/Omniladder/COSC390_Benchmarking
- Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., & Narasimhan, K. (2023). *Tree of Thoughts: Deliberate Problem Solving with Large Language Models*. arXiv preprint arXiv:2305.10601. https://doi.org/10.48550/arXiv.2305.10601
- Besta, M., Blach, N., Kubicek, A., Gerstenberger, R., Podstawski, M., Gianinazzi, L., Gajda, J., Lehmann, T., Niewiadomski, H., Nyczyk, P., & Hoefler, T. (2024). *Graph of Thoughts: Solving Elaborate Problems with Large Language Models*. arXiv preprint arXiv:2308.09687. https://doi.org/10.48550/arXiv.2308.09687
- princeton-nlp. (2023). *tree-of-thought-llm* [Source code]. GitHub. https://github.com/princeton-nlp/tree-of-thought-llm
- spcl. (2024). *graph-of-thoughts* [Source code]. GitHub. https://github.com/spcl/graph-of-thoughts
- run-llama. (2024). *llama_index* [Source code]. GitHub. https://github.com/run-llama/llama_index
- LlamaIndex Team. (2024). *LlamaIndex Documentation*. LlamaIndex. https://docs.llamaindex.ai/en/stable/