Trace-of-Thought Prompting: Investigating Prompt-Based Knowledge Distillation Through Question Decomposition

Tyler McDonald and **Ali Emami**
Brock University, St. Catharines, Canada
{tmcdonald3, aemami}@brocku.ca

Abstract

Knowledge distillation allows smaller neural networks to emulate the performance of larger, teacher models with reduced computational demands. Traditional methods for Large Language Models (LLMs) often necessitate extensive fine-tuning, which limits their accessibility. To address this, we introduce Trace-of-Thought Prompting, a novel framework designed to distill critical reasoning capabilities from large-scale teacher models (over 8 billion parameters) to small-scale student models (up to 8 billion parameters). This approach leverages problem decomposition to enhance interpretability and facilitate human-in-the-loop interventions. Empirical evaluations on the GSM8K and MATH datasets show that student models achieve accuracy gains of up to 113% on GSM8K and 20% on MATH, with significant improvements particularly notable in smaller models like Llama 2 and Zephyr. Our results suggest a promising pathway for open-source, small-scale models to eventually serve as both students and teachers, potentially reducing our reliance on large-scale, proprietary models. Our code, featuring data analytics and testing scripts, is provided here.

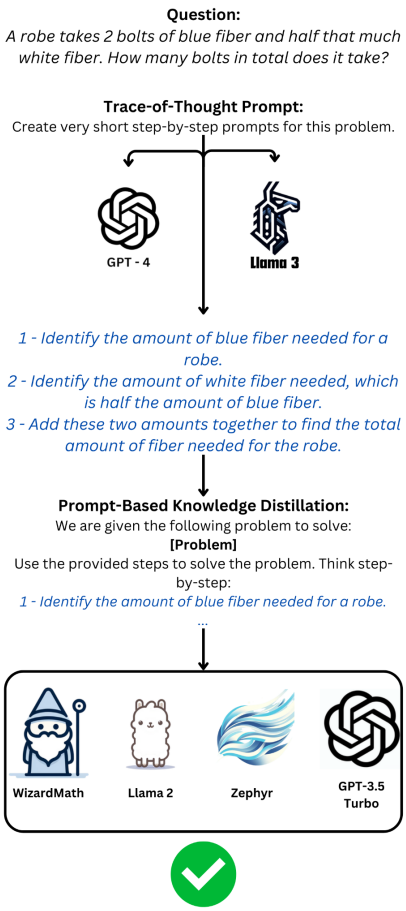

Figure 1: A Visual Depiction of our Trace-of-Thought prompting strategy on a GSM8K problem instance.

1 Introduction

Knowledge distillation, as initially proposed by Hinton et al. (2015), involves leveraging the outputs of larger neural networks as soft targets to train smaller, more efficient networks. This method, primarily applied to tasks like MNIST (LeCun et al., 1998) in computer vision, uses computationally heavy teacher models to facilitate equivalent reasoning capacities in smaller models, substantially reducing computational demands on the user. As the popularity of Large Language Models (LLMs) has surged, adaptations of this technique have been explored, particularly through fine-tuning based on the outputs of these large models. However, these adaptations often introduce a significant computational overhead and necessitate a deep understanding of machine learning, limiting their accessibility for average consumers (Xu et al., 2024; Gu et al., 2024; Liu et al., 2024; Zhong et al., 2024).

Concurrently, the rapid development of LLMs has been paralleled by innovations in prompt engineering—the strategic design of prompts to enhance reasoning and explore various problem-solving pathways (Sahoo et al., 2024; Chen et al., 2024a). Methods such as Chain-of-Thought Prompting and Self-Consistency have demon-

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 397–410

strated the potential of LLMs to engage in complex reasoning and provide novel solutions to challenging problems (Wei et al., 2023; Wang et al., 2023b; Yao et al., 2023; Wang et al., 2023a). Nevertheless, these approaches typically operate within a single contextual framework and rely heavily on the innate reasoning capabilities of models, often failing when applied to smaller, open-source variants (Touvron et al., 2023; Tunstall et al., 2023; Xu et al., 2023). This suggests a critical need for a more adaptable and scalable approach to knowledge distillation that can leverage the advances in prompt engineering for broader accessibility and effectiveness.

In response to this need, we explore the intersection of prompt engineering and knowledge distillation through a novel concept we term *prompt-based knowledge distillation*. This approach utilizes in-context learning (ICL) to emulate traditional distillation processes within the accessible framework of LLM prompting, mirroring the cognitive process of a student learning from a teacher (Brown et al., 2020). To implement this concept, we introduce *Trace-of-Thought Prompting*, a technique that decomposes complex arithmetic reasoning problems into manageable steps, facilitating the distillation of critical reasoning skills from large-scale models to their small-scale counterparts (see Figure 1). This strategy not only improves the performance of small-scale models but also demonstrates their potential to serve as effective teachers themselves.

Our contributions to this novel extension of knowledge distillation are threefold:

1. We propose *Trace-of-Thought Prompting*, a novel framework for prompt-based knowledge distillation. This approach allows knowledge transfer from large-scale models (greater than 8 billion parameters) to small-scale models (up to 8 billion parameters) through structured problem decomposition.

2. We demonstrate significant performance enhancements across two complex arithmetic reasoning datasets. By applying Trace-of-Thought Prompting, we improve the performance of small-scale models on the GSM8K dataset by 113% and on the MATH dataset by 20%. Our results also illustrate the effectiveness of small-scale models, like Llama 2 and Zephyr, in achieving performance gains that make them viable alternatives to their large-scale counterparts.

3. Our extended analyses demonstrate that Trace-of-Thought Prompting not only enhances quantitative performance metrics but also improves the transparency of the problem-solving process. This transparency allows for more effective human-in-the-loop interventions, where incorrect or suboptimal reasoning paths generated by the models can be identified and corrected before execution.

2 Related Work

Decomposed reasoning. Traditional question decomposition methods, including Plan & Solve Prompting and Progressive Hint Prompting, engage in single-context question decomposition, integrating a planning stage followed by an execution phase (Wang et al., 2023a; Press et al., 2023; Sun et al., 2023). More sophisticated recursive techniques, such as Least-to-Most Prompting, sequentially append results to enhance the context for subsequent prompts (Zhou et al., 2023; Dua et al., 2022; Khot et al., 2023; Zheng et al., 2023). These methodologies, however, face significant challenges: single-context systems fail to effectively leverage multiple models simultaneously, limiting flexibility and adaptability; recursive techniques, while intricate, hold the potential to lead to extended input sequences and excessive computational demands by virtue of their repetitive nature (Guo et al., 2024; Mohtashami et al., 2024; Juneja et al., 2024). Our Trace-of-Thought Prompting addresses these issues by facilitating dynamic, multi-model cooperation without the need for expansive input chains, streamlining the reasoning process across varied contexts.

Open-source language modeling. The rise of open-source models like WizardLM, Zephyr, and Llama has democratized access to language model customization and deployment (Xu et al., 2023; Touvron et al., 2023; Tunstall et al., 2023; Gunasekar et al., 2023; Team et al., 2024). Despite their accessibility, the teams behind these models report frequent deficiencies in complex reasoning tasks in small variants, underscoring a persistent correlation between model size and reasoning capabilities (Agrawal et al., 2024; Chen et al., 2024b; Zhang et al., 2024). Trace-of-Thought Prompting enhances these models' performance by distilling complex reasoning from larger models into manageable steps, effectively bridging the gap in reasoning prowess without extensive hardware de-

mands.

Tandem and Socratic reasoning. The exploration of collaborative problem-solving in model suites, such as Socratic Chain-of-Thought and Socratic Questioning, introduces novel ways to utilize multiple models in a cohesive manner (Shridhar et al., 2023; Qi et al., 2023; Chang, 2024; Zeng et al., 2022; Goel et al., 2024). However, these approaches encounter difficulties with managing large context sizes and reliance on fine-tuning (Li et al., 2024; Wang et al., 2024). Our work contributes to this area by implementing a structured approach that minimizes token bloat and fine-tuning dependency, offering a more efficient and scalable solution for collaborative reasoning within LLM environments.

3 Prompt-Based Knowledge Distillation

Traditional knowledge distillation, as originally proposed by Hinton et al. (2015), involves fine-tuning smaller neural networks on the soft outputs (logits) of larger, teacher networks. This transfer learning method enhances the smaller model's performance to emulate its larger counterpart, albeit with significantly reduced computational overhead. Despite its effectiveness, traditional knowledge distillation is resource-intensive, necessitating extensive computational efforts and substantial data, which limits its accessibility for average users.

In contrast, we introduce *prompt-based knowledge distillation*. This novel approach leverages in-context learning (ICL) to facilitate knowledge transfer without the extensive fine-tuning traditionally required. It conditions a small-scale student model on carefully crafted prompts derived from the large-scale teacher model, significantly reducing computational demands and enabling rapid adaptation to new tasks.

Consider the general framework for prompt-based knowledge distillation, where a teacher model $\mathcal{T}$ and a student model $\mathcal{S}$ interact. The teacher model processes an input question q to generate an informative prompt P, which encapsulates key insights or directions rather than explicit answers:

$$\mathcal{T}(q) \rightarrow P$$

The student model $\mathcal{S}$ then uses this prompt to infer the answer a, leveraging the distilled knowledge without direct output replication:

$$\mathcal{S}(P) \rightarrow a$$

Consider an educational scenario where a student model is required to solve geometry problems involving circle areas. For the problem "Calculate the area of a circle with a diameter of 10 cm," a large-scale teacher model could generate a prompt that distills essential concepts into several key points:

- Remember that the radius is half the diameter.
- Use the area formula for a circle: πr^2.
- Always include units in your answer (e.g., square cm).

This structured prompt guides the student model to focus on the fundamental mathematical relationships and proper problem-solving practices. By applying these principles, the student model calculates the radius as 5 cm and then uses the formula to determine the area as 25π square cm. This approach not only aids in solving the current problem but also reinforces good mathematical practices for future tasks.

4 Trace-of-Thought Prompting

Many problems in domains such as arithmetic reasoning can be broken down into intermediate steps that mimic the cognitive process of a human evaluator. Trace-of-Thought Prompting, an application of the prompt-based knowledge distillation framework introduced earlier, enhances models' problem-solving capabilities by breaking down such problems into simpler, actionable steps.

4.1 Formalization

We define a general language model $\mathcal{L}$ that processes an input $\mathcal{I}$ into an output $\mathcal{O}$:

$$\mathcal{L}(\mathcal{I}) \rightarrow \mathcal{O}$$

Assuming our input q is a problem that can be decomposed, we structure it into a sequence of interdependent steps:

$$q \rightarrow \{s_1, s_2, \ldots, s_n\}$$

The first step in Trace-of-Thought Prompting involves the decomposition of the problem into steps interpretable by a target model. The teacher model, $\mathcal{L}_T$, approximates the set of steps required to solve q:

$$\mathcal{L}_T(q) \xrightarrow{\approx} \{s_1, s_2, \ldots, s_n\}$$

Prompt Type	Template
Standard	"<question>."
Chain-of-Thought	"<question>. *Think step-by-step.*"
Plan & Solve	"<question>. *Let's first understand the problem and devise a plan to solve the problem. Then, let's carry out the plan and solve the problem step-by-step.*"
Trace-of-Thought - Delegation	"*Create step-by-step prompts for this problem:* <question>. *Format as a list of simple instructions to guide a student. Do not solve the problem.*"
Trace-of-Thought - Solution	"*First, carefully review this problem:* <question>. *Then, solve the problem using the provided steps as a plan, thinking step-by-step:* <steps>."

Table 1: Prompting templates used in experimental evaluation.

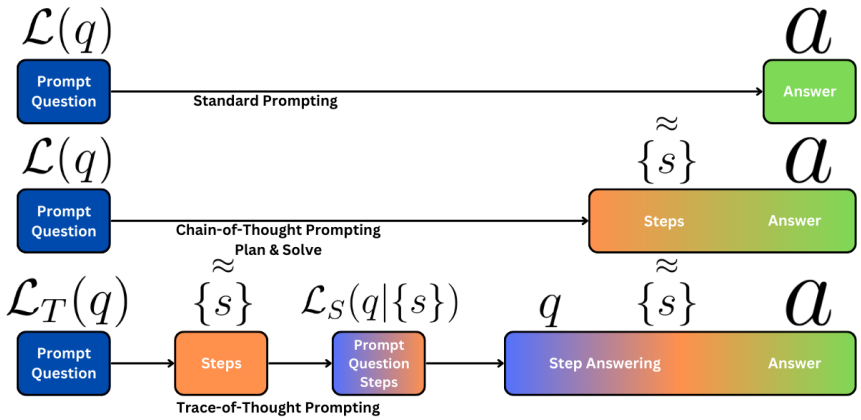

Figure 2: Visual depiction of the methods employed during experimentation. Trace-of-Thought provides a novel decomposition framework in a linear manner.

These steps are then used by the student model, $\mathcal{L}_S$, which is tasked with solving the original problem conditioned on the provided steps, aiming to generate the correct answer a:

$$\mathcal{L}_S(q|\{s\}) \rightarrow a$$

4.2 Practical Application Example

Consider the following problem q: "Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?"

During the distillation phase, the teacher model is prompted to consider this question as a framework for instruction, and to create simple question decompositions that can be passed to the student model alongside the original question:

> **Teacher Model – Distillation Phase:**
> `Create step-by-step prompts for the following problem:` q
> `Format as a list of simple instructions to guide a student. Do not solve the problem.`

Crucially, the teacher is instructed to not solve the input problem; instead, the output should consist solely of decomposed steps that aid in identifying strong reasoning pathways.

As a result, the teacher model might generate these steps:

- Identify April's sales.
- Calculate May's sales as half of April's.
- Add April's and May's sales to find the total.

Following the distillation phase, the student model receives both the input question and the generated decomposition prompts, and is instructed to think through these provided prompts step-by-step to ensure accuracy. At this point, the solution process utilizes traditional prompt engineering techniques to encourage the student to generate a high-quality answer via careful, transparent reasoning:

> **Student Model – Solution Phase:**
> `First, carefully review this problem:` q
> `Then, solve the problem using the provided steps as a plan, thinking step-by-step:` s

Model Name	Standard	Chain-of-Thought	Plan & Solve	Trace-of-Thought (GPT-4)	Trace-of-Thought (Llama 3)
GSM8K ($n = 200$)					
GPT-4	94.5	**95.5**	**95.5**	95.0	83.0
GPT-3.5-Turbo	75.5	73.5	74.5	**86.5**$^{\alpha}$	64.5
Llama 3 Instruct 8B	73.0	73.0	68.5	**88.0**$^{\alpha}$	63.5
WizardMath-7B	69.0	73.5	**82.5**	81.5	70.5
Llama 2 Chat 7B	22.0	23.5	23.0	**50.0**$^{\alpha}$	37.5$^{\alpha}$
Zephyr	26.0	23.5	30.0	**55.0**$^{\alpha}$	43.0
MATH ($n = 200$)					
GPT-4	57.5	66.0	**75.0**	68.0	55.0
GPT-3.5-Turbo	46.5	52.0	**56.0**	**56.0**	40.5
Llama 3 Instruct 8B	30.5	35.5	30.0	**41.0**	23.0
WizardMath-7B	**44.5**	33.5	37.0	42.5	30.5
Llama 2 Chat 7B	6.5	7.0	5.0	**8.0**	6.5
Zephyr	7.0	12.0	9.0	13.5	**14.5**

Table 2: Evaluation results for both GSM8K and MATH, $n = 200$. α denotes results where Trace-of-Thought's gain over the highest alternative was significant at $\alpha = 0.05$ (see Appendix Tables 7, 8, 9 and 10).

```
Student Output:
1. April sales: 48 clips.
2. May sales: 24 clips.
3. Total sales: 48 + 24 = 72
clips.
```

The student model uses the steps provided to compute the final answer: 72 clips. This approach not only ensures the student model understands the process of solving the problem but also maintains the structure of the reasoning path laid out by the teacher model.

Table 1 showcases the exact text necessary for the delegation and solution prompts, where the question and steps are interpolated as needed. A visual comparison with popular prompting approaches is provided in Figure 2.

5 Experimental Setup

5.1 Benchmarks

To evaluate the effectiveness of Trace-of-Thought in a practical environment, we select two arithmetic reasoning datasets of varying difficulty:

1. **GSM8K** (Cobbe et al., 2021) — GSM8K is a dataset of 8 thousand grade school level arithmetic reasoning problems, with a focus on simple problems that require some level of variable identification and decomposed reasoning.

2. **MATH** (Li et al., 2023) — MATH is a dataset of 50 thousand synthetically generated mathematical reasoning problems; MATH primarily focuses on a mix of simple and difficult arithmetic reasoning problems, with extended domains such as complex numbers, geometric reasoning, calculus, and functions.

In order to appropriately evaluate performance on these datasets, we sample $n = 200$ examples from each dataset, using each of the prompts in Table 1 on a suite of models.

5.2 Prompting Approaches

To evaluate each sampled problem, we employ a suite of popular prompting approaches in the literature:

1. **Zero-Shot Standard Prompting** — where each sampled question makes up the sole input to the model, with no in-context information provided.

2. **Zero-Shot Chain-of-Thought Prompting** — where each sampled question is appended with instructions to "think step-by-step" as proposed in Wei et al. (2023) and Kojima et al. (2023).

3. **Zero-Shot Plan & Solve** — where models are instructed to process the question, devise a plan of action, and solve that plan step-by-step prior to the question being provided, as proposed in Wang et al. (2023a).

4. **Zero-Shot Trace-of-Thought Prompting** — where a model is first instructed to decompose a problem into steps, before those steps are passed to another model instance for solution. Two variants are employed: **GPT-4** as a large-scale teacher model, and **Llama 3 Instruct 8B** as a small-scale teacher model.[1]

[1]Note that while Tree of Thoughts and Least-to-Most Prompting also fall under decomposition frameworks, their recursive nature is often difficult to properly emulate and does not align with the linear approaches suggested herein.

5.3 Evaluation

After a question is fully solved, the inputs, outputs, and provided dataset label are written to a file for human evaluation. The full set of testing data, comprised of 12 thousand total samples, is then human annotated by the authors, collectively familiar with all mathematical concepts leveraged by either dataset. Answers are annotated with a 1 if the output matches the provided label, and a 0 otherwise. The resulting score, given out of 200, is then tabulated as a percentage accuracy score for reporting.[2]

6 Results

Table 2 reports the accuracy results of each model and prompting approach on both datasets; the uppermost partition corresponds to results on GSM8K, while the lower corresponds to results for MATH. It demonstrates that Trace-of-Thought prompting outperforms many of the recent prompting approaches in both datasets.

6.1 Large-Scale Teachers - GPT-4

When applying GPT-4 as a large-scale teacher, on 58.3% of testing suites across both datasets, large-scale Trace-of-Thought generates results with the highest absolute accuracy scores. While some gains are slightly more nuanced — such as those observed when applied to GPT-4 on MATH — many small-scale models see strong accuracy gains when endowed with critical reasoning distilled from GPT-4. In the greatest of such cases, Llama 2's performance on GSM8K sees a rise of 27% absolute accuracy from 23% to 50% when queried using Trace-of-Thought.

6.2 Small-Scale Teachers - Llama 3

While Llama 3 as a teacher model does not encourage such gains as GPT-4, we observe that traditionally less performant models — such as Llama 2 and Zephyr — benefit strongly from distillation from a much smaller model than that of a large-scale teacher. On GSM8K, and with just a 14% size difference between teacher and student, we observe absolute accuracy gains of 14.5% and 13% on Llama 2 and Zephyr respectively.

[2]The data files used for evaluation, along with the scripts for analysis, will be made available in a public repository linked in the Abstract. Comprehensive documentation will accompany the data to assist researchers in replicating and extending the study.

Model	$\bar{x}_{HPA}$	Trace-of-Thought	% Gain
GSM8K ($n = 200$)			
GPT-4	95.5	95.0	-0.52
GPT-3.5-Turbo	75.5	86.5	**14.57**
Llama 3 Instruct 8B	73.0	88.0	**20.55**
WizardMath-7B	82.5	81.5	-1.21
Llama 2 Chat 7B	23.5	50.0	**112.77**
Zephyr-7B	30.0	55.0	**83.30**
MATH ($n = 200$)			
GPT-4	75.0	68.0	**3.03**
GPT-3.5-Turbo	56.0	56.0	0.00
Llama 3 Instruct 8B	35.5	41.0	**15.49**
WizardMath-7B	44.5	42.5	-4.49
Llama 2 Chat 7B	7.0	8.0	**14.29**
Zephyr-7B	12.0	13.5	**12.50**

Table 3: Relative gain on highest performing alternative approach ($\bar{x}_{HPA}$) - **large-scale teacher** (GPT-4).

Model	$\bar{x}_{HPA}$	Trace-of-Thought	% Gain
GSM8K ($n = 200$)			
GPT-4	95.5	83.0	-13.09
GPT-3.5-Turbo	75.5	64.5	-14.57
Llama 3 Instruct 8B	73.0	63.5	-13.01
WizardMath-7B	82.5	70.5	-14.55
Llama 2 Chat 7B	23.5	37.5	**59.57**
Zephyr-7B	30.0	43.0	**43.33**
MATH ($n = 200$)			
GPT-4	75.0	55.0	-26.67
GPT-3.5-Turbo	56.0	40.5	-27.68
Llama 3 Instruct 8B	35.5	23.0	-35.21
WizardMath-7B	44.5	30.5	-31.46
Llama 2 Chat 7B	7.0	6.5	-7.14
Zephyr-7B	12.0	14.5	**20.83**

Table 4: Relative gain on highest performing alternative approach ($\bar{x}_{HPA}$) - **small-scale teacher** (Llama 3).

6.3 Relative Accuracy Changes

There is an inherent issue of scale when considering performance improvements or drawbacks of using Trace-of-Thought. Tables 3 and 4 show the relative gains or losses of Trace-of-Thought on each student model at both teacher model scales.

A majority of models benefit from large-scale distillation with GPT-4; gains tend to be slightly more incremental on other higher-resource models (GPT-4, GPT-3.5-Turbo) or domain fine-tuned (WizardMath-7B) models, while gains are more notable on models of less scale and ability, occasionally nearing or exceeding 100%.

6.4 Effects of Scale on Performance

Figures 3 and 4 report relative gains sorted by average of absolute performance, or the average of a model's performance on every approach for each dataset. Models near the bottom of these figures tend to perform worse on a testing suite of multiple approaches; models near the top tend to perform

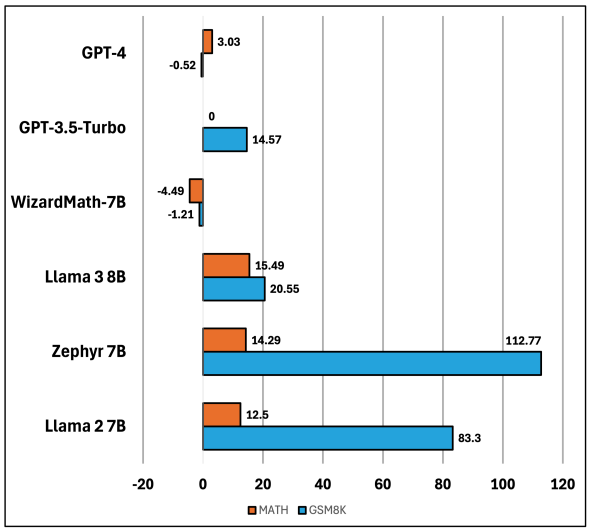

Figure 3: Relative accuracy changes with Trace-of-Thought (**large-scale**) visualized, in order of absolute performance.

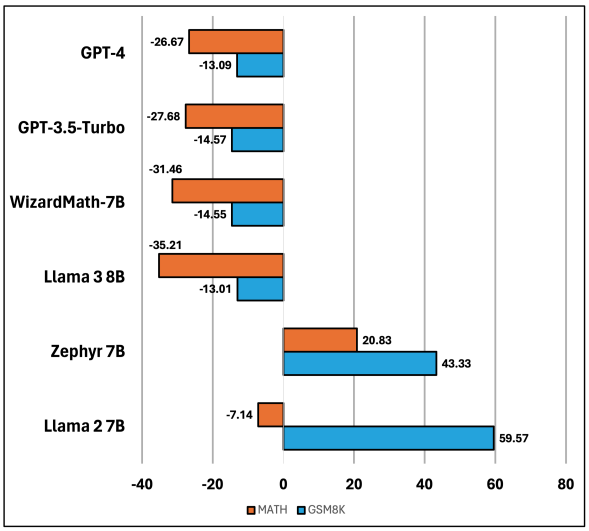

Figure 4: Relative accuracy changes with Trace-of-Thought (**small-scale**) visualized, in order of absolute performance.

well irrespective of the employed method.

When employing a large-scale teacher such as GPT-4, we observe near-universal gains through the usage of Trace-of-Thought Prompting. As scale continues to decrease, we observe consequently larger benefits, with under-performing, small-scale models such as Llama 2 7B and Zephyr-7B seeing the largest improvements of 83.3% and 112.77% respectively. Notably, this trend holds when employing a small-scale teacher such as Llama 3 8B; while performance gains do not distill *upwards* in scale, they do become most apparent in smaller models with historically less performance. In similar fashion, Llama 2 7B and Zephyr-7B see performance improvements of 59.57% and 43.33% respectively, demonstrating the potential of small-scale teacher models when paired with student models of lesser scale. In other words, *the weaker the student model is in comparison to the teacher model, the more it benefits from Trace-of-Thought Prompting.*

7 Qualitative Analysis

7.1 Transparent Reasoning

Trace-of-Thought Prompting enhances transparency in reasoning processes compared to previous approaches, allowing for more effective human-in-the-loop or automated corrections before erroneous reasoning is executed. This capability is crucial, as highlighted by Wu et al. (2024), who discuss the challenges of interpretability and faithfulness in models that only allow post-hoc evaluations. Consider this example on GSM8K:

Input:
`If Marcy works for the same company for 40 years, she gets an annual pension of $50,000/year. Starting after 20 years, she becomes entitled to 5% of the value of the pension per year. If she quits after 30 years, what will her annual pension be?`

Output (Llama 3):
`Determine how many years Marcy worked before becoming entitled to additional pension benefits (20 years).`

`*Calculate the value of Marcy's annual pension after 20 years. This is her basic pension without the additional benefit.`
(erroneous)

In this GSM8K example, the Trace-of-Thought method allows for identifying and correcting the misinterpretation of the pension calculation rules before finalizing the answer, which is an advantage in educational and practical applications.

7.2 Adaptive Reasoning

Trace-of-Thought enhances the adaptive reasoning capabilities of models, allowing them to adjust their problem-solving strategies based on the provided prompts. This adaptability reflects human cognitive processes and is vital for handling complex,

Error Type	Input	Output
Llama 3 - Incorrect Distillation	Carla is downloading a 200 GB file. Normally she can download 2 GB/minute, but 40% of the way through the download, Windows forces a restart to install updates, which takes 20 minutes. **Then Carla has to restart the download from the beginning.** How long does it take to download the file?	*Step 6: Calculate the new download time Calculate how long it will take Carla to **download the remaining 60%** of the file at her usual rate. **Result (Zephyr): *260 mins.** (Correct: 180)
GPT-4 - Incorrect Distillation	If Marcy works for the same company for 40 years, she gets an annual pension of $50,000/year. **Starting after 20 years, she becomes entitled to 5% of the value of the pension per year.** If she quits after 30 years, what will her annual pension be?	*3. **Calculate the base pension that Marcy is eligible for after 20 years.** This is because after 20 years, she starts accumulating an additional 5% per year. **Result (Llama 2): *$378,125** (Correct: $25,000)

Table 5: Teacher errors encountered when employing GPT-4 or Llama 3 as a teacher model.

Error Type	Input & Steps	Output
Zephyr - Incorrect Solution	James decides to run 3 sprints 3 times a week. He runs 60 meters each sprint. How many total meters does he run a week? ... 4. Identify the number of days James runs in a week. 5. **Multiply the total meters run each day by the number of days he runs each week to find the total meters run in a week.**	5. To find the total meters James runs in a week, we multiply the total meters he runs each day by the number of days he runs: 180 meters per day x 9 days = ***1620 meters in one week.** (Correct: 540m)
Llama 2 - Incorrect Solution	A candle melts by 2 centimeters every hour that it burns. How many centimeters shorter will a candle be after burning from 1:00 PM to 5:00 PM? ... 5. To find out how many centimeters shorter the candle will be, **multiply the total burning hours by the melt rate.**	5. To find out how many centimeters shorter the candle will be, multiply the total burning hours by the melt rate: 5 hours x 2 cm/hour = 10 cm. So, after burning from 1:00 PM to 5:00 PM, ***the candle will be 10 centimeters shorter.** (Correct: 8cm)

Table 6: Student errors encountered when employing Zephyr or Llama 2 as a student model.

real-world scenarios. Didolkar et al. (2024) highlights how this flexibility enhances models' ability to generalize and creatively apply knowledge. An example from GSM8K demonstrates this:

Input:
`Question: Consider the function` $g(x) = 3$. `Find` $g(2)$.

`Steps:`

`Review the function definition: **g(x) = 3**`

`Identify the target: **g(2)**`

`Substitute x with 2 in the function: **g(2) = ?**`

`Acknowledge the function's constant value: **? = 3**`

Output (GPT-4):
`Given` $g(x) = 3$, **`substituting any value for x results in 3.`** `Thus,` $g(2) = 3$.

In this case, the model efficiently applies its understanding that the output of a constant function is unchanging, thereby eliminating unnecessary computational steps. Moreover, Trace-of-Thought's flexible framework supports a form of interactive learning where models can both explore new problem-solving pathways and refine existing strategies.

7.3 Error Analysis

Table 5 illustrates examples of teacher errors, or errors in the distillation process; Table 6 subsequently demonstrates examples of student errors, or traditional reasoning flaws.

A minor drawback to Trace-of-Thought Prompting is the increased opportunity for knowledge pollution; in traditional approaches, incorrect solutions are contaminated by the single model responsible for solutions, while with Trace-of-Thought,

the teacher can pollute the downstream reasoning given to the student. Additionally, the student can respond adversely to the distilled reasoning, even if the reasoning provided is correct. This dual capacity for occasional wrongful reasoning encourages the selection of teacher models who respond well to the provided task, but ultimately does not discount the possibility of a student model that is traditionally unsuccessful on the same task. To aid in reducing troublesome distillations from the teacher model, various common strategies can be integrated in parallel with Trace-of-Thought, such as iterative verification prompts with Self-Consistency or Chain-of-Verification, or in-context learning given strong domain examples of high-quality question decomposition (Wang et al., 2023b; Dhuliawala et al., 2023; Brown et al., 2020).

8 Conclusion

This paper introduces a structured approach to prompt-based knowledge distillation, building on traditional methods to enhance accessibility and practicality for end-users. Our methodology, Trace-of-Thought Prompting, serves as a practical implementation of this framework, designed to facilitate problem decomposition and improve problem-solving capabilities in both large-scale and small-scale models. Through our experiments with various teacher model sizes, we have demonstrated how Trace-of-Thought can effectively leverage the knowledge distilled from both large and small models, improving reasoning capabilities in a variety of contexts. Our results show significant gains in model performance, especially in scenarios involving small-scale models, highlighting the potential of this approach to make AI more accessible and effective for a broader range of applications.

Limitations

Distillation of solution. While the Trace-of-Thought prompt is not intended to directly distill the solution, an exact study of the number of cases, if any, was not performed. As such, implementations of Trace-of-Thought should include a prompt tuning stage to ensure the teacher model is not strongly attending to solving the problem rather than distilling it further. The authors took proactive steps to disqualify answers that were directly distilling final results rather than instructive, guiding steps.

Recursive prompt study. Due to the computationally complex nature of implementing recursive methods such as Least-to-Most and Tree of Thoughts Prompting, there is a lack of comparison between the linear method of Trace-of-Thought and the similar recursive methods proposed in prior literature (Zhou et al., 2023; Yao et al., 2023). Future work should expand this testing battery to ensure an objective comparison between most prior literature and our implementation.

Restricted evaluation domain. Trace-of-Thought was designed primarily for use on arithmetic reasoning datasets; however, we have not tested its efficacy on various other domains. These domains may include abstract reasoning, commonsense reasoning, primarily linguistic datasets such as the Winograd Schema Challenge, among others (Clark et al., 2018; Srivastava et al., 2023; Wang et al., 2024; Sun and Emami, 2024). Further adaptations to the prompt structure may be necessary to fully adapt to these myriad tasks.

Restricted model scale. While small-scale models around the 7 billion parameter landmark have been evaluated, well-optimized small language models like Phi — between 1 and 3 billion parameters — have not been evaluated as students or teachers (Gunasekar et al., 2023). It remains to fully be seen if the trends in scale and performance hold across very small models such as these.

Improving students and teachers. Though Trace-of-Thought aided in performance gains, many performance losses observed on small-scale teachers are likely rectified through the improvement of instructions delegated through a fine-tuning process. The omission of fine-tuning in this paper was to provide an authentic comparison to the consumer language modelling experience, but further work should investigate the effects of fine-tuning a teacher model on a set of high-quality instructions and distillation practices (Ballout et al., 2024).

Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada and by the New Frontiers in Research Fund. Tyler McDonald is supported by the Natural Sciences and Engineering Research Council of Canada's Undergraduate Student Research Award.

References

Palaash Agrawal, Shavak Vasania, and Cheston Tan. 2024. Can llms perform structured graph reasoning?

Mohamad Ballout, Ulf Krumnack, Gunther Heidemann, and Kai-Uwe Kuehnberger. 2024. Show me how it's done: The role of explanations in fine-tuning language models.

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.

Edward Y. Chang. 2024. Socrasynth: Multi-llm reasoning with conditional statistics.

Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. 2024a. Unleashing the potential of prompt engineering: a comprehensive review.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024b. Self-play fine-tuning converts weak language models to strong language models.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models.

Aniket Didolkar, Anirudh Goyal, Nan Rosemary Ke, Siyuan Guo, Michal Valko, Timothy Lillicrap, Danilo Rezende, Yoshua Bengio, Michael Mozer, and Sanjeev Arora. 2024. Metacognitive capabilities of llms: An exploration in mathematical problem solving.

Dheeru Dua, Shivanshu Gupta, Sameer Singh, and Matt Gardner. 2022. Successive prompting for decomposing complex questions.

Anmol Goel, Nico Daheim, and Iryna Gurevych. 2024. Socratic reasoning improves positive text rewriting.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. Minillm: Knowledge distillation of large language models.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are all you need.

Xudong Guo, Kaixuan Huang, Jiale Liu, Wenhui Fan, Natalia Vélez, Qingyun Wu, Huazheng Wang, Thomas L. Griffiths, and Mengdi Wang. 2024. Embodied llm agents learn to cooperate in organized teams.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Gurusha Juneja, Subhabrata Dutta, and Tanmoy Chakraborty. 2024. LM^2: A simple society of language models solves complex reasoning.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. Decomposed prompting: A modular approach for solving complex tasks.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for "mind" exploration of large language model society.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024. Long-context llms struggle with long in-context learning.

Chengyuan Liu, Yangyang Kang, Fubang Zhao, Kun Kuang, Zhuoren Jiang, Changlong Sun, and Fei Wu. 2024. Evolving knowledge distillation with large language models and active learning.

Amirkeivan Mohtashami, Florian Hartmann, Sian Gooding, Lukas Zilka, Matt Sharifi, and Blaise Aguera y Arcas. 2024. Social learning: Towards collaborative learning with large language models.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models.

Jingyuan Qi, Zhiyang Xu, Ying Shen, Minqian Liu, Di Jin, Qifan Wang, and Lifu Huang. 2023. The art of socratic questioning: Recursive thinking with large language models.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. A systematic survey of prompt engineering in large language models: Techniques and applications.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2023. Distilling reasoning capabilities into smaller language models.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, César Ferri Ramírez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodola, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A. Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Wang, Gonzalo Jaimovitch-López, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Shevlin, Hinrich Schütze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B. Simon, James Koppel, James Zheng, James Zou, Jan Kocoń, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh D. Dhole, Kevin Gimpel, Kevin Omondi, Kory Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Colón, Luke Metz, Lütfi Kerem Şenel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramírez Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L. Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael A. Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Swędrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan A. Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Miłkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan LeBras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima, Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi,

Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven T. Piantadosi, Stuart M. Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsu Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models.

Jing Han Sun and Ali Emami. 2024. Evograd: A dynamic take on the winograd schema challenge with human adversaries.

Simeng Sun, Yang Liu, Shuohang Wang, Chenguang Zhu, and Mohit Iyyer. 2023. Pearl: Prompting large language models to plan and execute actions over long documents.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology.

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.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment.

Cangqing Wang, Yutian Yang, Ruisi Li, Dan Sun, Ruicong Cai, Yuzhu Zhang, Chengqian Fu, and Lillian Floyd. 2024. Adapting llms for efficient context processing through soft prompt compression.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Yexin Wu, Zhuosheng Zhang, and Hai Zhao. 2024. Mitigating misleading chain-of-thought reasoning with selective filtering.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A survey on knowledge distillation of large language models.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models.

Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Pete

Florence. 2022. Socratic models: Composing zero-shot multimodal reasoning with language.

Yunxiang Zhang, Muhammad Khalifa, Lajanugen Logeswaran, Jaekyeom Kim, Moontae Lee, Honglak Lee, and Lu Wang. 2024. Small language models need strong verifiers to self-correct reasoning.

Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo Li, and Yu Li. 2023. Progressive-hint prompting improves reasoning in large language models.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2024. Panda: Prompt transfer meets knowledge distillation for efficient model adaptation.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-most prompting enables complex reasoning in large language models.

Appendix

Two Sample Z-Test for Proportions - significant gains are bolded, and their significance level is put in brackets.

Model	Trace-of-Thought($\bar{x}_{ToT}$)	Highest Performing Alternative ($\bar{x}_{HPA}$)	z	p
GPT-4	95	95.5	-0.1662	—
GPT-3.5-Turbo	86.5	75.5	1.9827	**0.04770** ($p < 0.05$)
Llama 3 8B	88	73	2.6771	**0.00736** ($p < 0.01$)
WizardMath-7B	81.5	82.5	-0.1841	—
Llama 2 7B Chat	50	23.5	3.8866	**0.00001** ($p < 0.01$)
Zephyr-7B	55	30	3.576	**0.00034** ($p < 0.01$)

Table 7: Comparison of **large-scale** Trace-of-Thought performance against highest performing alternatives on the GSM8K dataset using two sample Z-test for proportions, $\alpha = 0.05$. Only scenarios with positive Z (gains) are reported.

Two Sample Z-Test for Proportions - significant gains are bolded, and their significance level is put in brackets.

Model	Trace-of-Thought($\bar{x}_{ToT}$)	Highest Performing Alternative ($\bar{x}_{HPA}$)	z	p
GPT-4	68	75	-1.0965	—
GPT-3.5-Turbo	56	56	0.0000	—
Llama 3 8B	41	35.5	0.8002	0.42372
WizardMath-7B	42.5	44.5	-0.2853	—
Llama 2-7B Chat	8	7	0.2685	0.78716
Zephyr-7B	13.5	12	0.3180	0.74896

Table 8: Comparison of **large-scale** Trace-of-Thought performance against highest performing alternatives on the MATH dataset using two sample Z-test for proportions, $\alpha = 0.05$. Only scenarios with positive Z (gains) are reported.

Two Sample Z-Test for Proportions - significant gains are bolded, and their significance level is put in brackets.

Model	Trace-of-Thought($\bar{x}_{ToT}$)	Highest Performing Alternative ($\bar{x}_{HPA}$)	z	p
GPT-4	83	95.5	-2.8536	—
GPT-3.5-Turbo	64.5	75.5	-1.6973	—
Llama 3 8B	63.5	73	-1.4431	—
WizardMath-7B	70.5	82.5	-2.0013	—
Llama 2 7B Chat	37.5	23.5	2.1502	**0.03156** ($p < 0.05$)
Zephyr-7B	43	30	1.9094	0.05614

Table 9: Comparison of **small-scale** Trace-of-Thought performance against highest performing alternatives on the GSM8K dataset using two sample Z-test for proportions, $\alpha = 0.05$. Only scenarios with positive Z (gains) are reported.

Two Sample Z-Test for Proportions - significant gains are bolded, and their significance level is put in brackets.

Model	Trace-of-Thought($\bar{x}_{ToT}$)	Highest Performing Alternative ($\bar{x}_{HPA}$)	z	p
GPT-4	55	75	-2.9650	—
GPT-3.5-Turbo	40.5	56	-2.1934	—
Llama 3 8B	23	35.5	-1.9430	—
WizardMath-7B	30.5	44.5	-2.0448	—
Llama 2-7B Chat	6.5	7	-0.1409	—
Zephyr-7B	14.5	12	0.5214	0.60306

Table 10: Comparison of **small-scale** Trace-of-Thought performance against highest performing alternatives on the MATH dataset using two sample Z-test for proportions, $\alpha = 0.05$. Only scenarios with positive Z (gains) are reported.