Can Large Language Models Reason About Goal-Oriented Tasks?

Filippos Bellos Yayuan Li Wuao Liu Jason J. Corso
University of Michigan, Ann Arbor, Michigan, USA
{fbellos,yayuanli,wuaoliu,jjcorso}@umich.edu

Abstract

Most adults can complete a sequence of steps to achieve a certain goal, such as making a sandwich or repairing a bicycle tire. In completing these goal-oriented tasks, or simply tasks in this paper, one must use sequential reasoning to understand the relationship between the sequence of steps and the goal. LLMs have shown impressive capabilities across various natural language understanding tasks. However, prior work has mainly focused on logical reasoning tasks (e.g. arithmetic, commonsense QA); how well LLMs can perform on more complex reasoning tasks like sequential reasoning is not clear. In this paper, we address this gap and conduct a comprehensive evaluation of how well LLMs are able to conduct this reasoning for tasks and how they scale w.r.t multiple dimensions(e.g. adaptive prompting strategies, number of in-context examples, varying complexity of the sequential task). Our findings reveal that while Chain of Thought (CoT) prompting can significantly enhance LLMs' sequential reasoning in certain scenarios, it can also be detrimental in others, whereas Tree of Thoughts (ToT) reasoning is less effective for this type of task. Additionally, we discover that an increase in model size or in-context examples does not consistently lead to improved performance.

1 Introduction

Large Language Models (LLMs) have transformed natural language processing (NLP), achieving groundbreaking performance across an array of tasks, primarily due to their capacity for (in-context) zero-shot and few-shot learning (Brown et al., 2020; Chowdhery et al., 2022; Vaswani et al., 2017). This prowess in task adaptation arises from their ability to “prompt"—essentially conditioning the models on limited examples or explicit task descriptions, and responding appropriately (Liu et al., 2021). The potential for models to adapt to tasks with limited to no exposure, especially without requiring extensive fine-tuning, is a testament to their potential and may be a step towards artificial general intelligence (Goertzel, 2014).

The ability to logical reasoning is one of the most intriguing capabilities of LLMs, which has been explored in various studies, including the evaluation of their grasp of commonsense knowledge (Davison et al., 2019; Liu et al., 2020; Ma et al., 2021; Niu et al., 2021; Zhou et al., 2020). Although their performance on intuitive and single-step tasks is exemplary, their efficacy on tasks requiring multi-step reasoning, particularly tasks that simulate human system 2[1] cognitive functions, has remained a challenge (Xu et al., 2023; Stanovich and West, 2000; Rae et al., 2021). This aspect of reasoning is vital, especially for goal-oriented tasks where the order and sequence in which actions are taken is crucial to the successful completion of the task.

Yet, in goal-oriented tasks, understanding and reasoning about a sequence of steps is critical. A disruption in the order of these steps can help, complicate or even nullify the task's objective. For example, in an effort to minimize speed in a certain task, such as preparing a soup in the kitchen, one must consider whether reordering certain steps is acceptable or by doing so the recipe (the goal) would be damaged. In the soup-making example, this could mean measuring, chopping and doing all preparation work—*mise-en-place*—before any cooking actually begins, which, oddly enough few recipes actually include as an explicit step but seems to not only speed up the overall cooking experience but lead to fewer later-step errors that would have otherwise resulted from inadequate inter-step time.

We are hence drawn to consider how well the recent advances in LLMs translate to the System 2-

[1] The term system 2 cognitive functions was coined by Kahneman (2011) and refers to the slow, analytical, reasoning-oriented thought processes, which are in contrast to system 1 cognitive functions that are instantaneous, subconscious reactions to stimuli.

Proceedings of the First edition of the Workshop on the Scaling Behavior of Large Language Models, pages 24–34
March 22, 2024 ©2024 Association for Computational Linguistics

type of reasoning, which we call *sequential reasoning*, necessary working with goal-oriented tasks.

Recent innovations, like the Chain of Thought prompting (CoT) (Wei et al., 2022; Wang et al., 2022), provide a promising solution to this reasoning challenge. Instead of relying on standard question-answer exchanges, CoT feeds LLMs with sequential reasoning examples, facilitating the model to map out a logical reasoning path. Alongside CoT there is an emerging technique known as Tree of Thoughts (ToT) prompting (Yao et al., 2023). ToT extends CoT's linear reasoning by allowing LLMs to explore multiple reasoning paths simultaneously, forming a tree of potential thoughts. This approach enables deliberate planning and exploration in problem-solving, where each thought is generated or solved independently. Moreover, there is an emerging interest in their inherent zero-shot reasoning skills (Brown et al., 2020). Novel approaches, such as Zero-shot-CoT (Liu et al., 2021), have demonstrated that by simply prompting models with an instruction like "Let's think step by step", LLMs can autonomously derive a plausible reasoning pathway and arrive at logical conclusions. Such findings not only underline the untapped potential of LLMs but also underscore their ability to mimic higher-level cognitive functions like generic logical reasoning (Chollet, 2019).

This is the first study that pushes this inquiry further, to evaluate LLMs' potential as logical reasoners for goal-oriented tasks, and investigate if the aforementioned claims for enhanced capability under certain prompting strategies hold true when used under the framework of *sequential reasoning*. Using adapted versions of the YouCook2 dataset (Zhou et al., 2018) and the CrossTask dataset (Zhukov et al., 2019) with varied sequence permutations, we probe the extent to which LLMs can discern and reason about the logical continuity of steps, especially when disruptions in their order are introduced (Fig. 1).

2 Methodology

Sequential tasks can be largely divided based on their properties, complexity, and dependence on previous steps. In this study, we focus on goal-oriented tasks - tasks that are directed towards achieving a particular objective, often encapsulated within a sequence of actions that must be executed in a specific order.

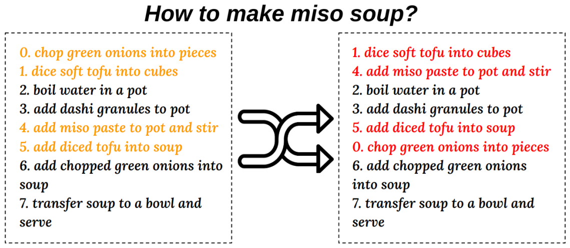

How to make miso soup?
0. chop green onions into pieces
1. dice soft tofu into cubes
2. boil water in a pot
3. add dashi granules to pot
4. add miso paste to pot and stir
5. add diced tofu into soup
6. add chopped green onions into soup
7. transfer soup to a bowl and serve
1. dice soft tofu into cubes
4. add miso paste to pot and stir
2. boil water in a pot
3. add dashi granules to pot
5. add diced tofu into soup
0. chop green onions into pieces
6. add chopped green onions into soup
7. transfer soup to a bowl and serve

Figure 1: **Illustration showcasing a permuted goal-oriented task**, specifically for preparing miso soup. On the left, the original recipe sequence is displayed, and on the right, the same recipe steps are shown in a permuted order. The example is from the YouCook2 dataset.

2.1 Properties of Goal-Oriented Tasks

Goal-oriented tasks share the following identifying properties.

Sequential Nature These tasks have steps; the steps are executed in a sequence. Although in practice two steps can be conducted at the same time—for example, two cooks in the kitchen can simultaneously measure out different ingredients—we assume only one step can be executed at one time. Steps may be repeated. For example, when preparing a peanut-butter-and-jelly sandwich, one must clean the knife after the peanut butter and then again clean it after the jelly.

Atomicity Each task in the sequence is atomic in nature, i.e., it represents a single, indivisable action. For instance, in cooking, "chopping an onion" could be considered an atomic action. The resolution of this atomicity is arbitrary and set by the experiment engineers or the dataset creators; we do not study the semantics of task-step resolution in this paper.

Dependency Later tasks in the sequence often depend on the completion and correctness of earlier tasks. For example, you cannot bake a cake without first mixing the ingredients.

Variability in Completeness While some steps are absolutely crucial, others might offer some leniency in terms of order or even necessity.

These properties yield the following situations regarding the success or failure to achieve the goal of a task. There is one or more prescribed ordering of the steps that are likely to lead to success; when one executes each step properly, it is expected to yield a successful outcome. We call this a "likely-success". However, one may still have not achieved the goal if certain steps are improperly executed. For the $N!$ possible orderings of tasks with N steps, a subset lead to a likely-success and the rest lead to failure.

2.2 Dataset Manipulation

The sequence in which goal-oriented tasks are carried out is pivotal. Yet, available goal-oriented datasets like YouCook2 (Zhou et al., 2018), HowTo100M (Miech et al., 2019) and COIN (Tang et al., 2019) do not contain permutations of task-steps that lead to failure; after all, they are instructional goal-oriented datasets. Therefore, for the sake of our study, we augment existing instructional, goal-oriented datasets to deliberately violate this order by introducing step permutations of different ratios, namely $1/2$ and $1/3$. By permutation ratio, we mean the ratio of the steps whose order has been modified.

Each of these permutations serves to disrupt the inherent flow of the goal-oriented task, leading to possible errors or alternative paths to reaching the goal.

We work with two datasets in this study, YouCook2 (Zhou et al., 2018) and CrossTask (Zhukov et al., 2019). We selected these two for their rich content that captures the complexity and sequential nature of goal-oriented tasks. We adapted a subset of these two datasets using a two-step process to optimally evaluate how disruptions in sequence can influence the outcome of these goal-oriented tasks and how LLMs can reason about this task structure. More details are in section 3.1.

2.3 Analysis Framework

Building on the goal-oriented task principles, our methodology critically assesses the capability of LLMs to reason about perturbed sequences. Acknowledging the atomicity of task steps and their inherent dependencies, we designed a set of prompts. When presented alongside permuted task sequences, these prompts task the LLMs with discerning the logical progression and determining the viability of the altered sequence.

To formulate our study, we present the two main analytical dimensions that our work is based on:

Assessment of Stepwise Transitions Our objective is to ascertain the proficiency of LLMs in understanding the logical coherence of task steps, even when perturbed.

Below we provide the input provided to the models, as well as the output that we expect.

<input>: Original goal-oriented task and its shuffled counterpart.

<output>: Step to step transition categorization into three types: (1) Correct: Step transitions with steps that retain their original sequential position; (2) Mistake: Disrupted sequences where the transition between the steps lacks logical or temporal coherence; (3) Variation: Step transitions that, despite being out of their original order, still maintain a logical flow that could conceivably be followed without detriment to the task.

Determining Task Viability On a macro scale, we aim to analyze the overall viability of the shuffled task. This entails identifying critical junctures, termed "Breaking Points", where modifications in sequence jeopardize the successful completion of a given task.

Below we provide the input provided to the models, as well as the output that we expect.

<input>: Original goal-oriented task and its shuffled counterpart.

<output>: Step transition that "breaks" the recipe.

In future sections we refer to the Assessment of Stepwise Transitions task as Task A and to the Determining Task Viability task as Task B.

Our prompt reasoning selection rationale is devised to span the entire logical reasoning spectrum, ensuring an in-depth and multi-faceted assessment of how LLMs understand goal-oriented tasks, and how they scale under different strategies.

2.4 Reasoning Strategies

We analyze model performance over three main pillars of in-context reasoning: Standard, Chain of Thought (CoT) and Tree of Thought (ToT) prompting.

For Standard Prompting, we directly ask for an answer. Specifically, we prompt with a question alone or a question and one or two $\langle$input, output$\rangle$ exemplars to potentially solve our task through direct explicit "reasoning".

For CoT Prompting, we provide zero, one or two examples of “chain of thought”, which are intermediate natural language reasoning steps, in the prompt to LLMs. Specifically, for zero-shot prompting we follow Kojima et al. (2022) and simply prompt LLMs with the phrase “Let’s think step by step” after the input, in order to elicit reasoning without the need for few-shot demonstrations. For one-shot and two-shot CoT prompting, we replace

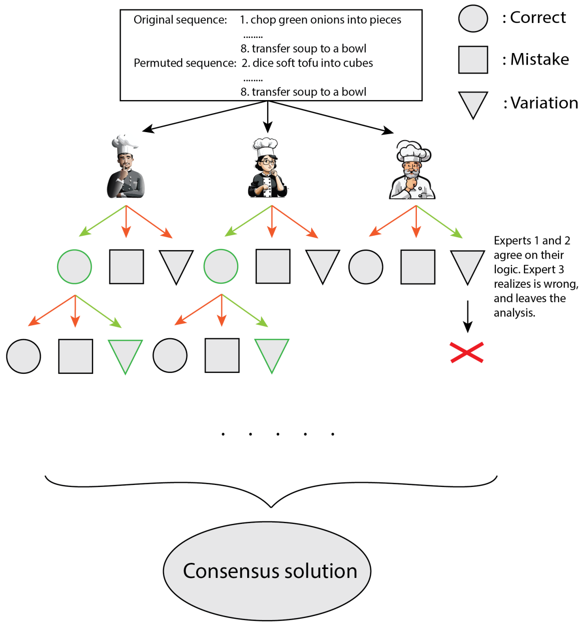

Original sequence: 1. chop green onions into pieces
....
8. transfer soup to a bowl
Permuted sequence: 2. dice soft tofu into cubes
....
8. transfer soup to a bowl
: Correct
: Mistake
: Variation
Experts 1 and 2 agree on their logic. Expert 3 realizes is wrong, and leaves the analysis.
Consensus solution

Figure 2: **ToT design for Task A.** We simulate the involvement of three experts analyzing our goal-oriented tasks, where each one explores at most 3 × N solution paths. Green arrows indicate paths chosen by an expert on each step transition. Red arrows indicate the other two possible paths not chosen by the expert for each step transition.

⟨input, output⟩ demonstrations with ⟨input, chain of thought, output⟩ triples.

To incorporate ToT in our study, we developed intricate prompts that simulate the involvement of three experts analyzing our goal-oriented tasks, such as evaluating the logical sequence of culinary steps in a shuffled recipe (as shown in Fig. 2). Each expert deliberately plans and reasons over the given task independently, exploring different solution paths. In the end, all experts reach a consensus solution.

In Task A, the experts deliberate after evaluating each step transition. If an expert finds their analysis to be incorrect, they withdraw from the discussion. After thoroughly analyzing and reasoning through the entire task's sequence, all experts agree on a final consensus solution. Specifically, as shown in Fig. 2, for each transition between steps, provided they have not exited the discussion, each expert explores 3 solution paths individually, one for each possible label "Correct", "Mistake", "Variation". This results in a total of 3 × N potential solution paths, with N representing the number of step transitions.

In Task B, a similar approach is followed, but here each expert is asked to reason over the whole task sequence, exploring individually N solution paths (worst-case). The ToT prompting arguably takes the form of self-consistency CoT here, since although the experts are prompted to reason step by step to find the breaking point, they follow single chain reasoning instead of a tree. Nevertheless, we will continue referring to it as ToT for Task B as well.

3 Experiments

3.1 Datasets

We use two datasets for our analysis. Both are goal-oriented datasets, primarily instructional datasets.

The first dataset is **YouCook2** (Zhou et al., 2018), a large-scale video dataset focusing on instructional cooking activities. Each one of 2000 videos is annotated with one of 89 recipe names and step-by-step instructions. Within the framework of this paper, they correspond to the concept of "goal" and "sequence" separately.

To adapt YouCook2 to our study, we further engage in a two-stage annotation process.

- First, we enhanced the annotation of several videos to include more nuanced labels that capture the complex progression of the recipes. Before this refinement, the videos typically had an average of 7.72 steps describing them. Post-refinement, this rose to an average of 12.06 steps. Our aim in this re-annotation was to segment the goal-oriented tasks such that each step represented a singular atomic action. This approach emphasizes the inherent sequential flow of these tasks.
- Second, we created two permuted versions of the re-annotated dataset (with ratios 1/2 and 1/3) and then performed a second round of annotations. Precisely, we annotated the stepwise transitions within the videos where we judged the correctness, variation or mistake in the logical and temporal order of the permuted version of the videos. These annotations assess the transition's fidelity to the original sequence and its logical and temporal validity.

The second dataset we use is **CrossTask** (Zhukov et al., 2019). It contains 18 *primary-tasks* and 65 *related- tasks*, a total of 4.7K videos. It covers a more diverse set of goal-oriented tasks, including tire changing, cooking, and furniture assembly. For our study:

- We evaluate only on the 18 primary task categories since they come with a full set annotation of temporal boundaries and step descriptions. The tasks have an average of 7.41 steps in sequence to fulfill a goal.

- Following the same procedure applied to the previous dataset, we create a permuted version of the CrossTask dataset (with ratio 1/2) and then proceed to annotate the stepwise transitions of each video based on their correctness, variation, or mistake.
- Noticeably, CrossTask has several tasks where repeated steps are performed to fulfill an ultimate goal. This detail adds an extra element of complexity that could affect the reasoning of LLMs about the logical continuity of steps.

The annotation process of stepwise transitions was carried out by 3 individuals to ensure accuracy and mitigate ambiguity.

The enhanced versions of these two datasets serve as the foundation for our experimental evaluation. In Table 1, we provide the statistics for both datasets.

Stepwise Transitions	**YouCook2**		**CrossTask**
	1/2	1/3	1/2
Correct	25.8%	51.0%	46.9%
Mistake	49.0%	29.6%	34.2%
Variation	25.2%	19.4%	18.9%

Table 1: **Stepwise transition statistics (%)** for our two datasets, YouCook2 (with 1/2 and 1/3 permutation ratio) and CrossTask.

3.2 Results

For our initial evaluation, we use OpenAI's GPT 3.5-turbo and GPT-4 models,.

The measure we choose to evaluate models is accuracy. Precisely, for Task A we evaluate the correct step transitions per goal-oriented task in our datasets and then we average over all of them: $\mathrm{Acc} = \frac{1}{N_{\text{tasks}}} \sum_{i=1}^{N_{\text{tasks}}} \mathrm{Acc}_i$, where $\mathrm{Acc}_i = \frac{\sum \text{Correct Step Transitions}}{N_{\text{Total Steps}}}$ is the accuracy for task i.

For Task B, we evaluate the if the breaking point of each task has been chosen correctly or not, and then we average over all tasks. We again calculate $\mathrm{Acc} = \frac{1}{N_{\text{tasks}}} \sum_{i=1}^{N_{\text{tasks}}} \mathrm{Acc}_i$, where now

$$\mathrm{Acc}_i = \begin{cases} 1 & \text{if breaking point for task } i \text{ is correct} \\ 0 & \text{otherwise} \end{cases}$$

for task i.

3.2.1 CoT and ToT Prompting Effect

To analyze the impact from applying CoT and ToT, we compute % point differences between CoT and Standard Prompting: $\mathrm{Acc}_{\mathrm{CoT}} - \mathrm{Acc}_{\mathrm{Standard}}$, as well as ToT and Standard Prompting: $\mathrm{Acc}_{\mathrm{ToT}} - \mathrm{Acc}_{\mathrm{Standard}}$. In our analysis, we use arrows to indicate ↑positive and ↓negative CoT and ToT effects.

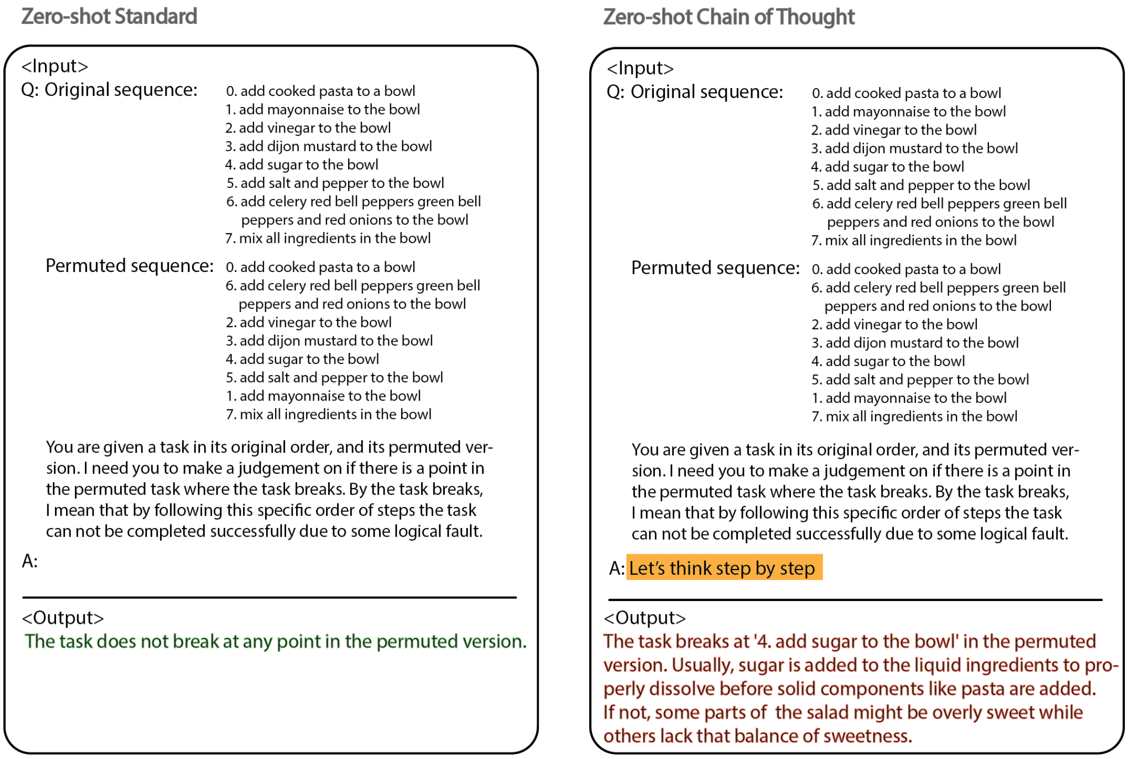
Task A Our experiments reveal that CoT and ToT prompting significantly enhances the capability of both GPT-4 and GPT-3.5-turbo models in reasoning over goal-oriented tasks, with CoT generally showing more consistent improvements (Table 2).

When evaluating GPT-4, both CoT and ToT show a consistent trend of improvement over standard prompting methods across different shot scenarios. For instance, in the YouCook2 dataset, zero-shot performance sees a notable increase with CoT (↑2.3%) and even more with ToT (↑3.3%). This pattern persists in one-shot and two-shot scenarios as well, though the benefits seem slightly more pronounced in the CoT approach. Interestingly, in some cases like the two-shot scenario in the CrossTask dataset, ToT shows a minor decrement (↓4.3%) compared to standard prompting.

GPT-3.5-turbo presents a different picture albeit with similar trends in terms of CoT and ToT improvements. Remarkably, GPT-3.5-turbo while able to understand the task under the zero-shot prompting strategy, when provided with examples under standard prompting, paradoxically it is unable to do so. This suggests that the provision of fully labeled examples of step transition sequences, rather than aiding the model, acts as a distractor, leading to repetitive, non-task-focused responses (e.g.repeating the examples in the answer). When prompted under CoT and ToT reasoning GPT-3.5-turbo was able to overcome this issue. Additionally, ToT seems to work exceptionally well for the CrossTask dataset but only similar to CoT for the YouCook2 dataset.

When using permutation ratio 1/3 the results are similar. However, the accuracy numbers are higher for all models, leading us to believe that LLMs can understand goal-oriented tasks better when there are less perturbations from the original sequence, and the logical coherence of the tasks is preserved.

Task B For this task, we specifically evaluate zero-shot capabilities, quantifying out-of-the-box performance. Models are sensitive to few-shot exemplars as seen from our results on Task A (table



Zero-shot Standard

<Input>
Q: Original sequence:
0. add cooked pasta to a bowl
1. add mayonnaise to the bowl
2. add vinegar to the bowl
3. add dijon mustard to the bowl
4. add sugar to the bowl
5. add salt and pepper to the bowl
6. add celery red bell peppers green bell peppers and red onions to the bowl
7. mix all ingredients in the bowl

Permuted sequence:
0. add cooked pasta to a bowl
6. add celery red bell peppers green bell peppers and red onions to the bowl
2. add vinegar to the bowl
3. add dijon mustard to the bowl
4. add sugar to the bowl
5. add salt and pepper to the bowl
1. add mayonnaise to the bowl
7. mix all ingredients in the bowl

You are given a task in its original order, and its permuted version. I need you to make a judgement on if there is a point in the permuted task where the task breaks. By the task breaks, I mean that by following this specific order of steps the task can not be completed successfully due to some logical fault.

A:

<Output>
The task does not break at any point in the permuted version.

Zero-shot Chain of Thought

<Input>
Q: Original sequence:
0. add cooked pasta to a bowl
1. add mayonnaise to the bowl
2. add vinegar to the bowl
3. add dijon mustard to the bowl
4. add sugar to the bowl
5. add salt and pepper to the bowl
6. add celery red bell peppers green bell peppers and red onions to the bowl
7. mix all ingredients in the bowl

Permuted sequence:
0. add cooked pasta to a bowl
6. add celery red bell peppers green bell peppers and red onions to the bowl
2. add vinegar to the bowl
3. add dijon mustard to the bowl
4. add sugar to the bowl
5. add salt and pepper to the bowl
1. add mayonnaise to the bowl
7. mix all ingredients in the bowl

You are given a task in its original order, and its permuted version. I need you to make a judgement on if there is a point in the permuted task where the task breaks. By the task breaks, I mean that by following this specific order of steps the task can not be completed successfully due to some logical fault.

A: Let's think step by step

<Output>
The task breaks at '4. add sugar to the bowl' in the permuted version. Usually, sugar is added to the liquid ingredients to properly dissolve before solid components like pasta are added. If not, some parts of the salad might be overly sweet while others lack that balance of sweetness.

Figure 3: **GPT-4's output when prompted with "Let's think step by step".** The model is distracted from the task's core objective—to evaluate the logical sequence of steps based on the original and permuted order. It states that sugar should be added before pasta, even though our recipe in its original order calls for adding pasta before sugar.

2) but also from the community (Zhao et al., 2021; Perez et al., 2021), so we want to avoid the variability that comes with them.

We observe a drop in performance when "Let's think step by step" prompting is applied. For GPT-4, when evaluating the YouCook2 dataset the accuracy declines from 64.6% to 54.2% (↓10.4%) for the 1/2 permutation, and from 72.9% to 57.1% (↓15.8%) for the 1/3 permutation. Similarly, in the CrossTask dataset with a 1/2 permutation ratio, GPT-4 experiences a decrease in performance, albeit a smaller one (↓1.9%). Likewise, GPT-3.5-turbo exhibits a decline, slightly more pronounced, in these scenarios.

The paradoxical phenomenon that arises in this task aligns with observations in the wider research community regarding the biases and background knowledge embedded in LLMs (Petroni et al., 2019). These biases can stem from the data on which they were trained, which can influence the performance of these models on tasks that require reasoning under narrow preconditions, like our permuted task sequence understanding. Essentially, the models may bring in their own "understanding" based on patterns they have learned, leading to accurate yet contextually irrelevant inferences, as seen in our experiment. For instance, GPT-4 provides factually correct statements regarding cooking procedures, such as sugar dissolving in liquid before mixing with solids to ensure flavor consistency (as illustrated in Fig. 3). However, it overlooks the task's core objective—to evaluate the logical sequence of steps based on the original and permuted order.

Looking at the ToT results we can see that having three paths with step-by-step zero-shot reasoning and taking the consensus solution from them causes a cascaded result and magnifies the zero-shot CoT issue. Each expert in their own path is carrying the model's bias in their decision attenuating the performance even further.

3.3 Scaling Behaviour

Chain of Thought (CoT) and Tree of Thought (ToT) are emergent behaviors typically associated with larger model scales. However, examining smaller models is crucial for understanding the scalability and potential limitations of these prompting strategies and their impact on sequential reasoning. We choose `Llama-2-13b-chat-hf` (Touvron

		GPT-4			GPT-3.5-turbo		
Dataset	N-shot	Standard	CoT	ToT	Standard	CoT	ToT
YouCook2	Zero-shot	62.2%	↑2.3 64.5%	↑3.3 65.5%	46.6%	↑0.2 46.8%	↑0.5 47.1%
	One-shot	66.0%	↑3.8 69.8%	↑0.7 66.7%	0.0%	↑47.0 47.0%	↑47.8 47.8%
	Two-shot	67.1%	↑3.3 70.4%	↓1.7 65.4%	0.0%	↑50.6 50.6%	↑46.8 46.8%
CrossTask	Zero-shot	69.5%	↑1.4 70.9%	↑0.4 69.9%	47.0%	↑0.3 47.3%	↑11.0 58.0%
	One-shot	71.3%	↑2.2 73.5%	↓1.4 69.9%	0.0%	↑48.4 48.4%	↑57.6 57.6%
	Two-shot	74.4%	↑3.2 77.6%	↓4.3 70.1%	0.0%	↑52.8 52.8%	↑57.9 57.9%

Table 2: **Performance comparison (%) of GPT-4 and GPT-3.5-turbo models under different reasoning strategies** across zero-shot, one-shot and two-shot scenarios for the YouCook2 (1/2 permutation ratio) and CrossTask datasets, for assessing stepwise transitions (Task A). Arrows indicate ↑positive or ↓negative impact of CoT and ToT compared to standard prompting.

Dataset	**Ratio**	**Standard**	**CoT**	**ToT**
		GPT-4		
YouCook2	1/2	64.6%	↓10.4 54.2%	↓14.3 50.3%
	1/3	72.9%	↓15.8 57.1%	↓26.9 46.0%
CrossTask	1/2	52.9%	↓1.9 51.0%	↑1.1 54.0%
		GPT-3.5-turbo		
YouCook2	1/2	20.8%	↓2.0 18.8%	↓2.8 18.0%
	1/3	36.7%	↓8.1 28.6%	↓19.6 17.1%
CrossTask	1/2	23.5%	↓3.9 19.6%	↓3.3 20.2%

Table 3: **Performance comparison (%) of GPT-4 and GPT-3.5-turbo models under standard, CoT and ToT zero-shot prompting for determining overall task viability (Task B)** on the YouCook2 dataset with 1/2 and 1/3 permutation ratios and the CrossTask dataset with a 1/2 permutation ratio. Arrows indicate ↑positive or ↓negative impact of CoT and ToT compared to standard prompting.

et al., 2023) which we will refer to as Llama-2-13b and zephyr-7b-beta (Tunstall et al., 2023) which we will refer to as Zephyr-7B-β. Llama-2-13b is the medium sized open source Language Model of its family of models and ideal size-wise for our scaling experiments. Zephyr-7B-β is even smaller, and was selected, over other models of the same size (like Llama-2-7b), to evaluate the performance of models trained using knowledge distillation techniques, where a smaller "student" model is trained based on the patterns learned by a larger "teacher" model. While distillation has been shown to improve smaller models, a gap compared to teacher models often still exists. Assessing an open distilled model allows us to directly test if the reported performance gains (Tunstall et al., 2023) hold across complex reasoning tasks.

We focus on the zero-shot scenario to avoid variability in experiments, and assess scalability patterns more reliably.

Task A For all datasets we observe that performance increases monotonically across scale (Fig. 5), with the exception of Zephyr-7B-β which outperforms the larger Llama-2-13b across different conditions. We hesitate to claim a "U-shaped" scalability pattern despite Zephyr-7B-β having fewer parameters than Llama-2-13b, as its training involves a larger model as a teacher, complicating direct comparisons based solely on parameter count. However, the strong performance of Zephyr-7B-β indicates that with proper training techniques, even relatively small models can achieve competitive results on complex reasoning tasks.

As far as scaling w.r.t prompting strategies, the analysis of the performance between CoT and ToT compared to the standard reasoning approach reveals a generally positive impact across models and datasets, with some exceptions.

In the YouCook2-1/2 dataset, both CoT and ToT techniques generally improve performance across all models. Notably, under ToT, GPT-4 shows a significant improvement with an increase of ↑3.3% points. Similarly, in CoT, Zephyr-7B-β and GPT-4 both exhibit an increase of ↑2.3% points each, indicating a consistent positive impact of these reasoning techniques.

Moving to the YouCook2-1/3 dataset, the trend largely continues. Under CoT, GPT-4 again demonstrates an increase, this time of ↑1.8% points. However, a slight deviation is observed with Llama-2-13b, which shows a small decrease of ↓0.6% points under CoT. Despite this, the overall trend remains positive. Interestingly, in the ToT approach, GPT-4 experiences a marginal decrease of ↓0.5% points, suggesting a more nuanced interaction in this par-

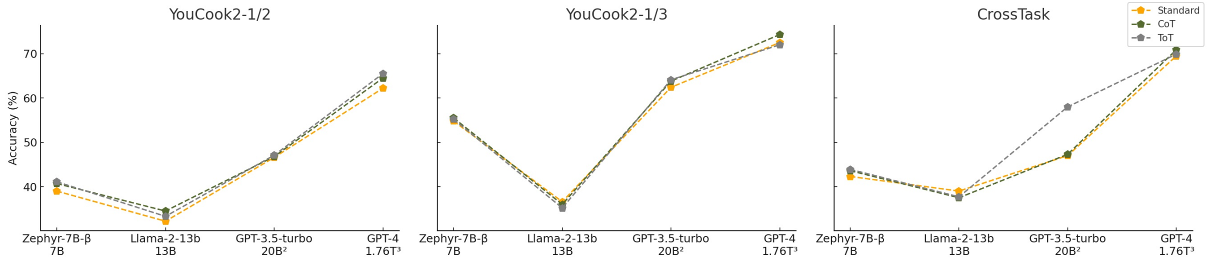


Figure 4: **Scaling Results for Task A** across models of different parameters for our benchmark datasets. Monotonic scaling behaviour is observed, even though Zephyr-7b-β outperforms Llama-2-13b in most cases.

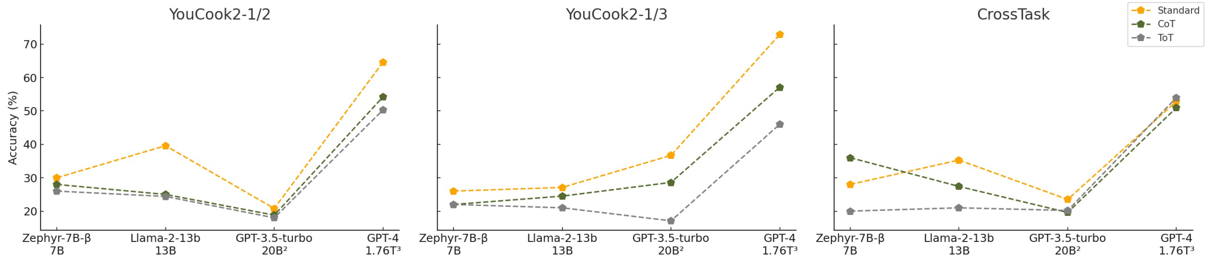


Figure 5: **Scaling Results for Task B** across models of different parameters for our benchmark datasets. "U-shaped" scaling behaviour is observed, as Zephyr-7B-β and Llama-2-13b outperform GPT-3.5-turbo.

ticular dataset.

The CrossTask dataset further illustrates the generally positive impact of CoT and ToT, with a standout increase in GPT-3.5-turbo's performance under ToT, showing a substantial improvement of ↑11.0% points. This is a significant observation, highlighting a particularly effective synergy between the ToT technique and the GPT-3.5-turbo model in this context. On the other hand, Llama-2-13b shows a decrease in both CoT (↓1.5% points) and ToT (↓1.3% points), marking it as an exception to the generally positive trend.

Overall, these findings suggest that while CoT and ToT reasoning techniques generally lead to improved performance over the standard approach, the extent of this improvement and its consistency can vary depending on the specific model and dataset.

Task B Across the YouCook2 and CrossTask datasets, we observe a "U-shaped" scalability pattern: where both Zephyr-7B-β and Llama-2-13b despite having significantly fewer parameters perform better than their larger counterpart, until GPT-4 overakes them in performance, indicating a critical threshold of model scale. In the CrossTask dataset, Zephyr-7B-β and Llama-2-13b, outperform GPT-3.5-turbo in both zero-shot standard (by ↑4.5% and ↑11.8% respectively) and zero-shot CoT prompting (by ↑16.4% and ↑7.8% respectively). For the YouCook2 dataset and the 1/2 condition, Zephyr-7B-β and Llama-2-13b outperform GPT-3.5-turbo in zero-shot standard (by ↑9.2% and ↑18.8% respectively) and CoT prompting (by ↑9.2% and ↑6.2% respectively). However, in the 1/3 condition, Zephyr-7B-β and Llama-2-13b underperform compared to GPT-3.5-turbo in zero-shot standard (by ↓10.7% and ↓9.6% respectively) and zero-shot CoT prompting (by ↓6.6% and ↓4.1% respectively), while again showcasing superior performance for ToT prompting.

4 Conclusion

In this work, we adapted and utilized the YouCook2 and CrossTask goal-oriented datasets to contain varied levels of step sequence permutations in order to analyze how Large Language Models respond to disruptions of logical order. We discover that CoT prompting strategies can significantly augment models' sequential reasoning capacities in some cases. However, it also unexpectedly harms reasoning performance under certain conditions. Moreover, ToT reasoning approaches prove less effective on perturbed goal-oriented tasks, while increases in provided in-context examples seems to improve model outcomes, but not across all cases.

[2] This number is reported by Singh et al. (2023) but it is not confirmed.

[3] This number is rumored but not officially released.

We also discover a "U-shaped" scaling behaviour, where LLMs with significantly less parameters perform better than one of their larger counterpart, in one of our tasks.

In total, while recent strategies can bolster goal-oriented reasoning, the models seem to have a fragile understanding of the complex dependencies in multi-step procedures, frequently overlooking logical flaws in permuted sequences. However, performance gains under simpler permutations indicates reasoning capability may rapidly improve alongside advances in scale and prompting.

Our analysis provides a methodology for continued investigation as models evolve on this challenging reasoning frontier. This study contributes to a deeper understanding of the scalability and adaptability of LLMs in complex reasoning tasks.

5 Limitations

Systematically exploring more reasoning strategies Our work uses different reasoning strategies, adapted for our tasks. However, small variations to the prompt structure could yield dramatically different results. Structuring ToT differently is one direction that could be explored. For task B, we focus on the zero-shot CoT prompting structure inspired by Kojima et al. (2022), and its extension to ToT. We need to expand our efforts by considering more prompting dimensions like adding in context exemplars in order to fully understand the cause of the performance drop and observe if the pattern persists.

Limitations of Sequential Reasoning Benchmarks Benchmarks often have varied interpretations of bias, leading to inconsistent outcomes (Delobelle et al., 2022; Cao et al., 2022). We introduce 2 separate benchmarks and evaluate LLMs reasoning on goal-oriented tasks across them. We believe our refined annotations and careful selection of the datasets to adapt are enough to mitigate the flaws of each individual benchmark, However, it's essential to carefully consider the inherent limitations and specific objectives of each benchmark when analyzing the results.

6 Ethics

This work involves experimentation with Large Language Models (LLMs) on goal-oriented reasoning tasks. As with any research involving LLMs, there are important ethical considerations.

Bias and Fairness Benchmarks can have inherent biases which can propagate to model evaluations. We aimed to mitigate this by using multiple datasets, but underlying biases may still exist. More broadly, the goal-oriented datasets likely contain some societal biases and future work should examine the extent of this.

Broader Societal Impact LLMs have potential benefits but also risks if deployed improperly. Our work aims to critically analyze these models, but downstream applications should carefully assess societal impact. If deployed to provide sequential guidance in real-world assistive systems, the reliability and safety of goal-oriented models is of utmost importance. Understanding model capabilities and limitations is crucial for avoiding potential harms from erroneous system behaviors.

Throughout this work, we attempted to conduct rigorous scientific exploration to further knowledge and understanding around the reasoning robustness. We believe this has value for enabling responsible applications in future, but also that researchers have an ethical duty to acknowledge risks and unintended consequences as language models continue advancing.

7 Acknowledgements

This work has been supported by the Defense Advanced Research Projects Agency (DARPA) under Contract HR00112220003. The content of the information does not necessarily reflect the position of the Government, and no official endorsement should be inferred.

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. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.*

Yang Cao, Yada Pruksachatkun, Kai-Wei Chang, Rahul Gupta, Varun Kumar, Jwala Dhamala, and Aram Galstyan. 2022. On the intrinsic and extrinsic fairness evaluation metrics for contextualized language representations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 561–570, Dublin, Ireland. Association for Computational Linguistics.

François Chollet. 2019. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Joe Davison, Joshua Feldman, and Alexander M Rush. 2019. Commonsense knowledge mining from pretrained models. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 1173–1178.

Pieter Delobelle, Ewoenam Tokpo, Toon Calders, and Bettina Berendt. 2022. Measuring fairness with biased rulers: A comparative study on bias metrics for pre-trained language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1693–1706, Seattle, United States. Association for Computational Linguistics.

Ben Goertzel. 2014. Artificial general intelligence: concept, state of the art, and future prospects. *Journal of Artificial General Intelligence*, 5(1):1.

Daniel Kahneman. 2011. *Thinking, fast and slow*. Farrar, Straus and Giroux.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS 2022)*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Ye Liu, Tao Yang, Zeyu You, Wei Fan, and S Yu Philip. 2020. Commonsense evidence generation and injection in reading comprehension. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 61–73.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Satoru Ozaki, Eric Nyberg, and Alessandro Oltramari. 2021. Exploring strategies for generalizable commonsense reasoning with pre-trained models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5474–5483.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2630–2640.

Yilin Niu, Fei Huang, Jiaming Liang, Wenkai Chen, Xiaoyan Zhu, and Minlie Huang. 2021. A semantic-based method for unsupervised commonsense question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3037–3049.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2021. Scaling language models: Methods, analysis & insights from training gopher.

Mukul Singh, José Cambronero, Sumit Gulwani, Vu Le, Carina Negreanu, and Gust Verbruggen. 2023. Codefusion: A pre-trained diffusion model for code generation.

Keith E Stanovich and Richard F West. 2000. Individual differences in reasoning: Implications for the rationality debate? *Behavioral and brain sciences*, 23(5):645–665.

Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. 2019. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1207–1216.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in NeurIPS*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models.

Fangzhi Xu, Qika Lin, Jiawei Han, Tianzhe Zhao, Jun Liu, and Erik Cambria. 2023. Are large language models really good logical reasoners? a comprehensive evaluation and beyond.

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.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models.

L. Zhou, C. Xu, and J. J. Corso. 2018. Towards automatic learning of procedures from web instructional videos. In *Proceedings of AAAI Conference on Artificial Intelligence*.

Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. 2020. Evaluating commonsense in pretrained language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9733–9740.

Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. 2019. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3537–3545.