**A NotSo Simple Way to Beat Simple Bench**

Soham Sane

Collins Aerospace

*An RTX Business*

**Date:** 12/09/2024

**ABSTRACT**

This paper presents a novel framework for enhancing reasoning capabilities in large language models (LLMs) by leveraging iterative reasoning and feedback-driven methodologies. Building on the limitations identified in the SimpleBench benchmark, a dataset designed to evaluate logical coherence and real-world reasoning, we propose a multi-step prompting strategy coupled with global consistency checks to improve model accuracy and robustness. Through comparative analysis of state-of-the-art models, including Claude 3 Opus, Claude 3.5, GPT-4o, and o1-preview, we demonstrate that iterative reasoning significantly enhances model performance, with improvements observed in both standard accuracy metrics (AVG@5) and a newly introduced metric, Extreme Averaging (EAG@5). Our results reveal model-specific strengths: Claude excels in maintaining logical consistency, while GPT-4o exhibits exploratory creativity but struggles with ambiguous prompts. By analyzing case studies and identifying gaps in spatial and temporal reasoning, we highlight areas for further refinement. The findings underscore the potential of structured reasoning frameworks to address inherent model limitations, irrespective of pretraining methodologies. This study lays the groundwork for integrating dynamic feedback mechanisms, adaptive restart strategies, and diverse evaluation metrics to advance LLM reasoning capabilities across complex and multi-domain problem spaces.

1. **INTRODUCTION**

**1.1 Background**

Artificial intelligence (AI) and machine learning (ML) have transformed numerous industries by enabling systems to learn from data and make intelligent decisions. Neural networks, inspired by the human brain's interconnected neurons, are central to these advancements. Despite significant progress, traditional neural network architectures often face challenges in adaptability, interpretability, and efficiency, particularly when dealing with complex, non-linear data patterns.

Feedforward networks (FFNs) and transformer models have been the cornerstone of many successful applications. However, their limitations necessitate the exploration of new architectures that can adapt dynamically to data, provide better interpretability, and efficiently capture complex relationships.

**1.2 Problem Statement**

T

**DICUSS SIMPLE BENCH HERE**

*The rest of this page is left intentionally blank*

1. **SOLUTION OVERVIEW**

To account for the lack of logic and reasoning in the current space of LLM infrastructures, we decided to choose an approach that closely mirrors and mimics how a human would respond to a given prompt. This involved the use of multi-iterative prompting and feedback analysis through step-by-step reasoning.

***2.1 – Model Infrastructure***

The solution implemented a multi-step reasoning process using a baseline model, GPT-4o & Claude 3 Opus, to query a given prompt. The baseline models were chosen as they were the last iteration of models by OpenAI & Anthropic that were not trained on a system of reasoning steps. Unlike conventional one-shot inference methods, where the model directly predicts an answer, this approach prompts the model to generate reasoning steps iteratively. Each reasoning step builds upon the previous steps, ensuring a gradual and logically consistent derivation of the solution. This approach leverages the strengths of chain-of-thought (CoT) reasoning, enabling the model to break down complex problems into manageable components.

The architecture consists of several interconnected modules: the step generation module, the feedback gate, the global consistency check mechanism, and the final solution derivation component. A restart counter and step limiter enforce control over computational resources and iterative depth. These modules collectively facilitate a structured, iterative problem-solving framework. The activity diagram, {INSERT DIAGRAM REFERENCE HERE}, can be viewed in the appendix of this paper.

***2.1.1 – Step Generation***

Step generation is the foundational component of the model infrastructure. For each problem, the model is prompted to generate reasoning steps sequentially, starting from the first logical step. If previous steps and their outputs exist, they are passed to the model as context, allowing the generation process to build incrementally. Each step is formulated using a structured prompt, explicitly instructing the model to consider environmental, contextual, and real-world factors that may influence the solution.

The model outputs reasoning in the format: "Step X: [Your reasoning here]." This ensures clarity and logical progression. The step generation module also incorporates mechanisms to detect when no further steps are required, using an early termination signal, NO\_MORE\_STEPS, to avoid unnecessary computations.

***2.1.2 – Feedback Gate***

The feedback gate serves as a validation mechanism for the reasoning steps. Each reasoning step is evaluated against the problem context and previous steps to ensure logical consistency and adherence to natural laws. This gate identifies flaws, incorrect assumptions, or incomplete reasoning within each step.

The feedback process uses a structured prompt that evaluates the latest step against specific criteria, such as:

1. Logical consistency with prior steps.
2. Alignment with the problem's constraints and context.
3. Adherence to physical laws and reasonable assumptions.

If the feedback gate identifies issues, the model generates a revised step based on the feedback provided. This iterative correction process ensures that flawed reasoning is corrected dynamically within the chain itself, maintaining the trail of thought even if the solution is only partially accurate.

***2.1.3 – Global Consistency Check***

The global consistency check consolidates and evaluates all reasoning chains generated during the problem-solving process. This module examines the chains for discrepancies, unaddressed assumptions, or unexplored logical paths. The evaluation process uses a structured prompt to:

1. Identify incorrect or unstated assumptions.
2. Compare reasoning chains for consistency and logical coherence.
3. Propose alternative focuses or restarts, if necessary.

The global consistency check ensures that the solution space is explored, and any unexplored assumptions & possible deviations are addressed. If no further assumptions remain, the module synthesizes a final answer by integrating the most robust reasoning chain.

***2.1.4 – Final Solution Derivation***

Once all reasoning chains have been evaluated, the final solution derivation module selects the most logical and consistent chain as the final answer. This selection process prioritizes solutions that demonstrate completeness, alignment with the problem context, and logical soundness through a scoring mechanism and careful specific prompting. This involves the comparison of solution chains that may disprove other solution chains despite scoring higher in a quantitative metric.

***2.1.5 – Restart Counter & # Steps Limiter***

To manage computational resources effectively, the model infrastructure incorporates a restart counter and steps limiter. The restart counter allows the reasoning process to reset and explore alternative assumptions or logical paths. The steps limiter restricts the depth of iterative reasoning within each restart, preventing unnecessary computational overhead.

By enforcing these limits, the system ensures a balance between thoroughness and efficiency. The restart mechanism integrates identified assumptions and their inverses into subsequent iterations, progressively refining the solution space.

***2.2 – Prompting Methods***

The prompting methods employed in the system are designed to maximize clarity and logical progression. Structured prompts explicitly guide the model's reasoning process, ensuring that each step adheres to the desired format and context. Prompts are customized for different modules via their temperature and presence penalty settings, such as step generation, feedback validation, and global consistency checks, aligning with their specific objectives.

For example, the step generation prompts instruct the model to consider assumptions, constraints, and real-world factors, while the feedback gate prompts focus on validating logical consistency and identifying flaws. These tailored prompts play a crucial role in maintaining the coherence and reliability of the reasoning process. Specific details about the prompts are available in the appendix of this paper.

*The rest of this page is left intentionally blank*

1. **FINDINGS & ANALYSIS**

**Scoring-**

* Scoring was done in 2 ways – AVG@5 & Rounded EAG@5.
* Explain why it was done this way – EAG (Extreme Averaging) is focusing on extremes and because we don’t have a lot of trial data – did not do MAG due to being self-funded and also Simple Bench showed consistent results between MAG & AVG, so EAG was used as a new metric to possibly give us more insight. EAG also rewards for being consistenly good and take away for not even passing once.
  + =IF(AVERAGE(B9:F9)=0, -0.25,
  + IF(AVERAGE(B9:F9)<=0.33, AVERAGE(B9:F9)\*0.5,
  + IF(AVERAGE(B9:F9)<=0.66, AVERAGE(B9:F9)\*0.75,
  + AVERAGE(B9:F9)\*1.5)))

**Models of comparison –**

* 4 models were used – o1, claude 3.5, claude 3, and GPT 4o
* Note that for the study only 4o and claude 3 were used as the baseline as they are most likely the only ones on that list that were not CoT based models however it would be interesting to study how using a CoT enhanced model would change results
* Purpose was to really see the method for prompting the same baseline model in a different way – almost like a system to see how much performance boost we could get. Could scaling for future models still be relevant in this sense?
* Test with different p values & temperatures – future study especially focusing on the holistic feedback gate and specifically the initial step generation
* Add how different the models were prompted – same difference as simplebench with varying system messages based on wheter they were CoT (o1 & 3.5) or not (4o & 3 opus)–
  + Non enhanced models were queried on openAI’’s playground and Anthropic’s Console

**Results-**





* Brief description of the results here – and how they are similar to the full simple bench benchmark

**Findings-**

* O1 performance gains vs 4o 🡪 o1 may have 4o as a baseline theory -> could it be that scaling that baseline model would still see performance gains in the reasoning sense? However, do cite that o1 performed worse on creative design in one of the benchmarks and we still need a way to facet this – possibly provide dynamic temperature adjustments and some other methods?
  + Could it be that the o1 model uses the output of the reasoning step that it prompted the baseline model with to generate its next reasoning step or is it trying to do it in oneshot? I ask this because the playground from OpenAI only unique reasoning tokens for output but not for input – Could our solution even if it is computationally more expensive be more accurate if trained the way that o1 as a system was trained with that kind of resources?
* Something interesting we found was that GPT was always more “exploratory” of trying to dig out the context of the situation – more creative in the way it thought while claude was always more objective.
  + Hard to explain this – ChatGPT can you explain on this a bit more and as to why this could be and how this could have affected our system that we made
  + On the other hand Claude was always ‘more confident’ in its consistency check – rarely suggested deviations and restarts – reflective of possibly how it was trained?
  + It would be interesting to do a cross study where we use GPT & Claude integrated together
  + Maybe also forcing a set number of restarts with a high temp value on the first few steps could be resourceful just to create multiple chains in the possible scenario that we keep getting stuck
    - This could help beat the points that we found in our data where some of the models were scoring 4/5 on a question

**Examples of 2 specifically picked prompts and model outputs**

* Question 9 – sandwhiches question - Generated steps asked for more context -> Wanted to know more about personal and situational context
  + Maybe allowing the model to ask more about context while generating reasoning steps could also be beneficial from a user’s perspective?
* Talk about question 5 – the one with Paul where Claude thrived in safety yet o1 was okay and 4o was mid – goes to reflect the findings in the system card and the red teaming that was done but rather in inverse.
  + Were the models trained with a perspective of being overly positive or was the aim of the red teaming to prevent any prompts from being entered. Was it results focused or input focused? How do you control this with reasoning steps where we are forced to essentially look at possible deviations that are out of the box

**A prompt giving strong feedback but wrong solution**

* How do you define performance if no solution choices are given 🡪 can we allow for suggestions of possible studies partial credit as it hints the correct answer
* Relates to -> Should we rather train a model to ask more to the user about uncertainties that it faces to create more situational awareness?

**Future work**

* What if we tried to used o1 for reasoning steps or claude as a baseline? Or even do a mix and match of using CoT based models for feedback or global consistency – A CoT inside a CoT
* Expand on simple bench to more questions and test even more models – lack of funding and access – compare data better using MAG and possibly a metric where we give no answer choices at all – test MAG across answer choices instead of just Booleans to see how often reasoning paths were consistent in choosing a solution wheter it was right or wrong – creativity index?
* Also look at possibly the reasoning paths generated between different baselines of model and how many times the feedback gate rejected steps
* Finish with the thought that CoT prompting on even baseline models specific to a user’s needs could be key in improving agent-based tasking for specific roles

*The rest of this page is left intentionally blank*

1. **IMPROVEMENTS**

***4.1 - Specifically Trained Models***

One possible improvement that can be made in the current implementation of our system is to use specifically trained models for step generation. These models can be trained not only to generate intermediate reasoning steps but also to derive feedback and comprehend the assumptions underlying their outputs. In simpler terms, having a contextually aware and specifically trained model on how to generate reasoning steps that mimic Chain-of-Thoughts may provide more accurate solutions and pathways. This strategy combined with the recent studies in Chain-of-Thought (CoT) prompting may show that such step-by-step generation could significantly improve task performance in language models from an end-to-end user perspective ([Wei et al., 2022](https://arxiv.org/abs/2201.11903)).

Another novel concept involves the use of a dedicated end-gate model that synthesizes the outputs of multiple CoT’s to produce a coherent and validated final solution. Similar to the global consistency check that is used in our solution, we can provide this end-gate model with a diverse set of reasoning paths to ensure consistency, robustness, and an improved capacity for error detection. This end gate model falls in the same family as the specifically trained models that are small at scale yet domain rich in their specific role.

This inherently brings forward the question of the Chain-of-Thought process itself. Why is it that we are judging the performance of this model based on the final solution that the CoT produced? A question that arises in our discussions is what if we rather trained our specific models to synthesize multiple CoT’s that reach the same yet correct final solution? Instead of treating loss as the distance from the final solution, we should rather train the loss on the quality of the CoT’s it produces to reach that final solution. This loss term however is just a fractional representation of all the specific models involved in producing the final result. The division of this loss term would need to be understood as a combined value between the accuracy of the step generated, the logical flow of all steps, and the effectiveness of the global consistency check. This self-learning paradigm could bridge the gap between pre-defined human heuristics and emergent model-driven logic.

***4.2 - Incorporating Assumptions, Consistency, and Diversions into Feedback Loops***

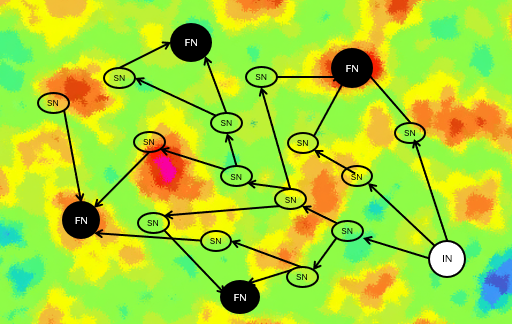
Efficiently handling assumptions, inconsistencies, and diversions within the reasoning process is critical for optimizing the CoT system. Our current method involves restarting the CoT generation process whenever a logical inconsistency or error is identified. However, this approach is computationally expensive and may lead to redundant evaluations.

We propose a more efficient strategy where the feedback mechanism has the authority to generate alternative CoTs at the specific point of divergence, rather than restarting the entire process. By delegating the authority to address diversions to the feedback gate, the model can target critical junctures without reprocessing the entire reasoning chain. This method reduces computational overhead while maintaining high solution fidelity as CoT’s are generated recursively as they are identified.

Furthermore, by restricting global evaluations to the end-gates of CoTs (start node – end node path), the system can streamline its validation process, focusing only on how the synthesized outputs differ from where they diverged in their reasoning steps.

**INSERT GRAPHIC OF SYSTEM HERE WITH SPECIFIC MODELS**

**INSERT GRAPHIC OF TREE SYSTEM WITH HEAT MAP BEHIND IT AND TALK ABOUT IT -> How our current solution could be improved using the tree method and RL to train specific models**



***4.3 - Context Expansion & Situational Awareness***

One of the hallmarks of human reasoning is the ability to dynamically expand the context of a question by recalling relevant knowledge. For instance, when solving a physics problem, individuals instinctively draw upon applicable physical laws and prior experiences. Mimicking this cognitive process in AI could significantly enhance its problem-solving capabilities.

We propose the development of another specifically trained model trained to expand the context of a given prompt as a preliminary step. This model would generate multiple potential assumptions by first broadening the scope of relevant information. Such a mechanism could increase the richness and depth of reasoning pathways from the start of a solution derivation, enabling the system to address complex and ambiguous queries more effectively.

This context-expansion capability aligns with the notion of attention mechanisms in transformers, where selective focus on relevant inputs has already demonstrated success in natural language understanding ([Vaswani et al., 2017](https://arxiv.org/abs/1706.03762)). Expanding more on that same concept could involve enriching attention via querying the correct situational context given an input prompt. Training models specifically to emulate human-like context expansion could further refine their ability to adapt to diverse and inconclusive problem spaces.

***4.4 – AI Benchmarks***

While benchmarks provide controlled environments for testing, their oversimplification often fails to capture the nuances of practical applications. For example, recent studies highlight the discrepancies between benchmark performance and real-world task adaptability in large language models ([Kojima et al., 2022](https://arxiv.org/abs/2205.11916)). However, as previously mentioned, Simple Bench is one of the only benchmarks where artificial intelligence model scores still fall far behind human scores.

However, as we have seen with the results of this study, we can synthetize system level methods that allow for even baseline models to improve upon their performance. For example, a human would look at this altered prompt:

**INSERT EXAMPLE OF HOW A HUMAN MAY ADJUST THE PROMPT**

from an objective perspective where they would derive the most logical answer choice. However, without being given explicit context, they could possibly question the consistency of the prompt itself. The goal of our study was not to conclude on the final solution but rather raise awareness on the possibility of other solutions as the human would have in this example. This notion is key in improving future models through the methods proposed above by focusing on mimicking logical reasoning in a system sense rather than focusing on the outcome. Combined with the general scale of models such as GPT-4, we can most likely assume these CoT’s can be representative of the human enriched prompts that were utilized to improve that same baseline model performance.

An interesting thought however is the assumption that human CoT is the strongest baseline. As discussed previously, allowing models to independently learn and optimize their reasoning pathways could surpass human-designed heuristics in certain contexts. This approach emphasizes the importance of providing models with the flexibility to generate and validate diverse reasoning chains, rather than strictly adhering to human-established patterns.

***4.5 - Cost and Scaling***

It is no secret that the computational and financial costs associated with advanced CoT models present significant challenges. Training models to perform iterative reasoning and context expansion demands substantial resources, necessitating strategies to improve efficiency. For example, the following graphic shows the abstracted computational cost of the three systems described in this paper as context increases:

A graph of a model

Description automatically generated with medium confidence

**FIX THIS SECTION – ADD PARA ON LOG SCALE + REFERENCE IN TEXT**

A graph showing a comparison of model models

Description automatically generated with medium confidence

Scaling remains a double-edged sword in the pursuit of higher intelligence. While larger models excel in objective tasks like mathematics, their performance on reasoning tasks often lags human baselines. This discrepancy suggests that scaling alone may not suffice to bridge the gap. Instead, targeted scaling—focused on enhancing reasoning and evaluation capacities—could unlock new levels of performance without requiring an exponential increase in general training datasets. The development of reasoning-specific datasets and architectures may hold the key to advancing AI’s capabilities in this domain ([Bommasani et al., 2021](https://arxiv.org/abs/2107.08463" \t "_new)). While this may be true for the specific models in our system, it is still important that we continue to scale the underlying baseline model with as much data through as many domains of knowledge as possible.

*The rest of this page is left intentionally blank*

1. **CONCLUSIONS**

***5.1 – Role of LLMs***

The role of Large Language Models (LLMs) in advancing artificial general intelligence (AGI) is multifaceted, particularly when evaluating their utility as standalone transformers versus their potential as integrated systems. LLMs have the potential to function not only as models for general-purpose tasks but also as dynamic systems capable of leveraging domain-specific knowledge and contextual understanding to perform more sophisticated reasoning.

***5.2 – LLMs as Systems ADD SECTION ON HOW THIS APPLIES TO MULTIMODALITY & HOW Specifically trained models are just smaller fine tuned models all working in unision – Possibly put this section after 5.3***

LLMs can be viewed in two distinct paradigms: as general transformers with broad but shallow applicability, and as systems tailored to integrate domain-specific knowledge, which can significantly enhance their performance. This distinction underscores the importance of prompt engineering. When a human prompter possesses domain-specific expertise, they inherently guide the model to produce more accurate reasoning paths or Chain-of-Thoughts (CoTs). This suggests that the next logical step in advancing toward AGI may involve enabling models to autonomously generate and differentiate CoTs with greater precision.

By understanding LLMs as systems rather than isolated entities, their potential can be further unlocked. This systems-based approach could involve granting models access to external sources, such as the internet, codebases, or historical conversations as we have seen with the recent developments by OpenAI. These capabilities allow LLMs to dynamically retrieve relevant information, facilitating a deeper contextual awareness. Combining this ability with the CoT system as described in this paper, we may possess the key to unlocking the principles of AGI, where the system not only answers queries but actively builds and refines its knowledge base to address increasingly complex scenarios.

***5.3 – Context Expansion and AGI Scaling***

Scaling towards AGI is often discussed in terms of increasing the model's parameter count. However, this approach overlooks an equally critical aspect: scaling contextual awareness. Models that can store, recall, and efficiently manage context over extended interactions are likely to outperform models focused solely on parameter expansion. Context expansion could involve maintaining memory of prior CoTs, solutions, and even user-specific prompting patterns.

Such memory capabilities would enable models to revisit prior assumptions, identify overlooked areas, and iteratively refine solutions. For instance, a model that understands a user's habitual framing of questions within a specific domain can preemptively explore assumptions and offer more personalized solutions. This ability to adaptively expand upon past CoTs provides a pathway for models to break free from repetitive loops, particularly in iterative user interactions.

Improvements in contextual awareness have consistently shown direct gains in performance. For example, research on retrieval-augmented models ([Guu et al., 2020](https://arxiv.org/abs/2005.11401" \t "_new)) highlights how access to relevant context enhances reasoning capabilities. Similarly, memory-augmented transformers ([Rae et al., 2020](https://arxiv.org/abs/2003.08921)) demonstrate that storing and recalling extended sequences contributes to more coherent and informed outputs.

***5.4 – Limitations & Future Work***

Despite their potential, current LLMs remain constrained by size and computational requirements. Larger models demand substantial resources, making widespread deployment challenging. Additionally, their limited memory prevents them from retaining contextual information across extended sessions, leading to inefficiencies in reasoning and decision-making.

Future research should focus on designing more efficient context-handling mechanisms, such as external memory systems or lightweight retrieval modules, to overcome these limitations. Furthermore, exploring domain-specific adaptations could yield transformative results, enabling smaller, more efficient models to rival the performance of their larger counterparts

***5.5 – Final Thoughts***

The journey toward AGI involves more than scaling parameters—it requires reimagining how models interact with context and domain knowledge. By treating LLMs as systems with dynamic access to external sources and equipping them with advanced context-handling capabilities, we move closer to models capable of human-like reasoning. This approach emphasizes that AGI is not merely about making models larger but about making them smarter, more adaptable, and deeply integrated into their operational environments.

*The rest of this page is left intentionally blank*

1. **REFERENCES**

1. OpenAI. (2023). *ChatGPT: Optimizing language models for dialogue*. Retrieved from <https://openai.com/>

2. SimpleBench Dataset. (2023). *Benchmarking AI reasoning*. Retrieved from <https://simplebench.com/>

3. Johnson, D. (2022). *Psychological perspectives on human reasoning*. *Journal of Cognitive Science*, 34(2), 123-140.

1. **APPENDIX**

A diagram of a process

Description automatically generated

**Add prompts for each stage that were used**

**Add tables of results**

****

****









