UC Merced

Proceedings of the Annual Meeting of the Cognitive Science Society

Title

Scaling up the think-aloud method

Permalink

https://escholarship.org/uc/item/8jc6r7h5

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 47(0)

Authors

Wurgaft, Daniel
Prystawski, Ben
Gandhi, Kanishk
et al.

Publication Date

2025

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at https://creativecommons.org/licenses/by/4.0/

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Scaling up the think-aloud method

Daniel Wurgaft*
Psychology
Stanford University
wurgaft@stanford.edu

Ben Prystawski*
Psychology
Stanford University
benpry@stanford.edu

Kanishk Gandhi
Computer Science
Stanford University

Cedegao E. Zhang
Brain and Cognitive Sciences
MIT

Joshua B. Tenenbaum
Brain and Cognitive Sciences
MIT

Noah D. Goodman
Psychology and Computer Science
Stanford University

Abstract

The think-aloud method, where participants voice their thoughts as they solve a task, is a valuable source of rich data about human reasoning processes. Yet, it has declined in popularity in contemporary cognitive science, largely because labor-intensive transcription and annotation preclude large sample sizes. Here, we develop methods to automate the transcription and annotation of verbal reports of reasoning using natural language processing tools, allowing for large-scale analysis of think-aloud data. In our study, 640 participants thought aloud while playing the Game of 24, a mathematical reasoning task. We automatically transcribed the recordings and coded the transcripts as search graphs, finding moderate inter-rater reliability with humans. We analyze these graphs and characterize consistency and variation in human reasoning traces. Our work demonstrates the value of think-aloud data at scale and serves as a proof of concept for the automated analysis of verbal reports.

Keywords: think-aloud; verbal protocol; reasoning; problem solving; language models

Introduction

Understanding how people reason is central to cognitive science, yet reasoning is notoriously elusive to study. Reasoning happens internally and unfolds over time, so studying it requires making inferences about the time course of a process we cannot observe directly. Throughout the field's history, cognitive scientists have developed methods to carefully infer thought processes from the data we *can* observe directly. One such method, which was prominent in the formative years of the field, is the *concurrent verbal protocol*, or *think-aloud*, method (Newell & Simon, 1972; Ericsson & Simon, 1993; Van Someren et al., 1994).

In a think-aloud study, experimenters present a participant with a problem and instruct them to say whatever comes to mind as they solve it. The experimenters then study the transcript of what the participant said to infer the process they went through on their way to the solution (Ericsson & Simon, 1993). Classic studies applied this method to investigate reasoning in varied settings such as chess (Newell & Simon, 1972), physics (Larkin et al., 1980), and insight problems (Kaplan & Simon, 1990). Think-aloud data informed the development of early computational models of cognition, including the General Problem Solver (Newell & Simon, 1961). The method provides rich data that elucidate the *process* of reasoning, in addition to its outcomes. Therefore, think-aloud studies let us analyze reasoning at the algorithmic level, in the framework of Marr (1982).

Despite its continued use in fields such as education and human-computer interaction (Van de Wiel, 2017; Hertzum, 2024), the think-aloud method has largely fallen out of favor in contemporary cognitive science. Traditional think-aloud studies relied on sample sizes of only one or few participants, since coding transcripts requires substantial training and time (Newell & Simon, 1972; Larkin et al., 1980). As the field embraced large-scale experiments to enhance the robustness of findings and explore individual differences (e.g., Hartshorne et al., 2018; Peterson et al., 2021), extensive manual coding of data became impractical.

The alternative to manual coding is an automated coding pipeline, which would allow the melding of process-level insights from think-aloud data with the large-scale experiments of contemporary cognitive science. Automatic coding of think-aloud data was originally proposed by Waterman & Newell (1971). However, the shortcomings of available artificial intelligence at the time limited the applicability of their system. Given recent developments in natural language processing (NLP)—particularly speech-to-text models (Radford et al., 2023) and large language models (LLMs) (Brown et al., 2020)—new tools are available to study verbal reports.

Therefore, we believe the time is right to revitalize the think-aloud method in cognitive science. Automated coding will enable researchers to process verbal reports from many participants, yielding data with both granularity and scale. Scale can also help mitigate concerns about the validity of think-aloud studies—specifically, that prompting participants to verbalize their thoughts may alter how they solve tasks (Wilson, 1994). While some studies support this claim (Russo et al., 1989), a more recent meta-analysis of 94 studies found no evidence that prompting to think aloud changes behavior (Fox et al., 2011). Still, large comparisons between reasoning behavior with and without verbalization are useful for understanding whether and how prompting to think aloud changes the distribution of final behavior.

Several recent studies have used NLP tools to analyze verbal reports in risky choice and memory tasks (Ostrovsky & Newell, 2024; Mechera-Ostrovsky et al., 2024; Xie et al., 2023, 2024a). These papers have provided valuable new ways to extract psychological information from unstructured audio and text. However, they focus on classifying whole utter-

*These authors contributed equally.

In D. Barner, N.R. Bramley, A. Ruggeri and C.M. Walker (Eds.), *Proceedings of the 47th Annual Conference of the Cognitive Science Society* ©2025 the author(s). This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY).

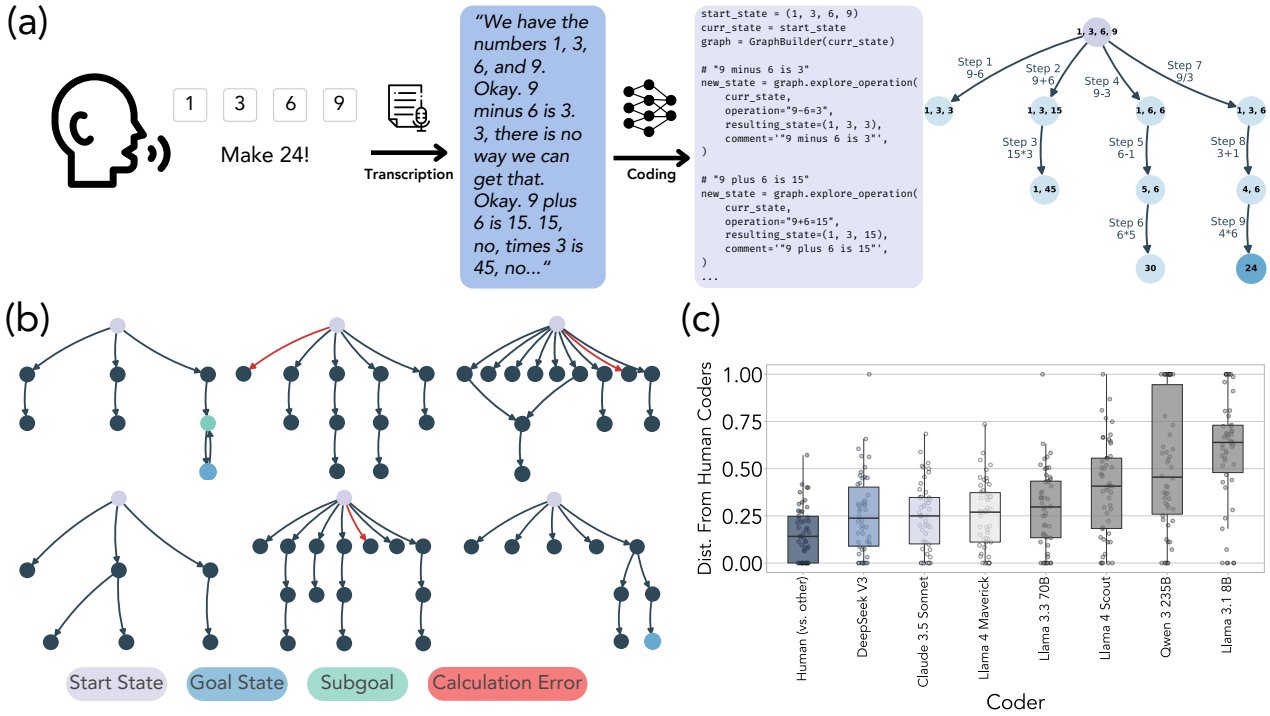


Figure 1: **Overview of the data processing pipeline.** (a) Participants solve the task while speaking aloud. We transcribe their data using a speech-to-text model and use an LLM to translate the transcripts into Python code that produces a graph representation of their search. (b) Example graphs for one problem in the dataset. (c) Normalized graph edit distances between the graphs produced by human and language model coders. Lower edit distances indicate more agreement.

ances in simple settings (Ostrovsky & Newell, 2024; Xie et al., 2023; Mechera-Ostrovsky et al., 2024). In contrast, many seminal think-aloud studies focused on producing *process-level* insights in reasoning tasks that allow us to understand the individual steps taken when reasoning, in the spirit of Newell & Simon (1961)'s early cognitive modeling work.

In this work, we develop and validate an automated coding pipeline for processing think-aloud data in the context of a mathematical problem-solving task. We use the Game of 24, a simple yet rich arithmetic task that is well-suited to the study of reasoning. This game, along with its generalized version (also known as Countdown), has been used to study the development of mathematical reasoning (Tong et al., 2014; van der Maas & Nyamsuren, 2017) and is a popular benchmark for evaluating LLMs' ability to reason (Yao et al., 2023; Besta et al., 2024; Gandhi et al., 2024). We analyze nearly 5,000 verbal traces of reasoning in what is, to the best of our knowledge, the largest think-aloud experiment ever run. Using a pipeline that codes participants' reasoning traces as search graphs, frontier LLMs display inter-rater reliability that is close to, but still less than, human-level. We then demonstrate the richness of think-aloud data by examining the generated search graphs and characterizing patterns in human reasoning traces. Thus, we argue this study offers a step towards realizing Waterman & Newell (1971)'s vision of automated process-level analysis of human reasoning.

Method

In our study, participants played the Game of 24, in which they are tasked with combining four starting numbers between 1 and 13 to make 24 using only basic arithmetic operations (addition, subtraction, multiplication and division). Participants in the think-aloud condition were asked to say all of their thoughts out loud as they solved the problem. Participants in the control condition were not prompted to verbalize. The preregistration for this experiment can be found here. Full code and data are available on GitHub.

Participants

We recruited 700 participants via Prolific, filtering for participants in the United States who were fluent in English, had at least a secondary education, and had an approval rate of at least 98%. 640 participants were assigned to the think-aloud condition and 60 were assigned to the control. Participants took a median of 34 minutes to complete the experiment. They received a base payment of $6.35 and earned a bonus of 10 cents for each problem they solved.

We excluded trials with transcripts that contained no information relevant to the task. If at least half of a participant's trials were excluded, we excluded the remaining trials as well. Of the 6400 trials we collected, 17 were excluded due to a technical error that corrupted the audio, 1240 were excluded due to a lack of relevant information, and 174 were excluded based on participant exclusions. We also excluded the 22 re-

maining trials from one of the problems, which was a repeat of a practice problem. This left us with 4,947 trials from 541 participants in the think-aloud condition.

Stimuli

We chose 200 problems of varying difficulty from 4nums.com, which contains crowdsourced data on the difficulties of Game of 24 problems. To ensure that difficulties were similar across condition groups, we ordered the 200 problems by solve rate (taken from 4nums), and divided them into 10 deciles by difficulty. We then randomly sampled one problem from each decile, yielding 20 groups of 10 problems.

Procedure

Participants in the think-aloud condition were randomly assigned to one of the 20 problem groups via counterbalancing, yielding 32 participants per group. We randomly assigned participants in the control condition to one of three problem groups, chosen from the 20 in the think-aloud condition. This gave us 20 participants per group.

Participants first read the instructions, which told them to try to say out loud everything that goes through their mind as they solve the task. Next, they completed two practice trials before continuing to the main experiment. The practice trials were the two easiest problems on 4nums. Each practice trial repeated until the participant answered correctly.

Participants completed 10 main trials. In each trial, they entered equations on the screen using buttons that corresponded to the four starting numbers and the arithmetic operations. If a participant submitted an incorrect solution, they received a message informing them it was incorrect and the trial continued. Trials ended after the participant entered a correct solution, or after 3 minutes elapsed.

We used an automatic voice detector to ensure that participants were verbalizing their thoughts consistently. Every 20 seconds, it instructed the participant to verbalize more if the average vocal activity at the human voice range was below a threshold. After each trial, participants were asked to rate the extent to which they were saying their thoughts out loud on a 5-point Likert scale. They were asked to verbalize more of their thoughts if they responded 3 or lower.

Analysis pipeline

After collecting data, we processed it using a pipeline that automatically transcribes recordings and turns them into graphs. This section describes the pipeline at a high level (see repository for additional details).

Transcribing recordings

We transcribed the voice recordings using OpenAI Whisper large-v3, a state-of-the-art open-weights automated transcription model (Radford et al., 2023). We used the hyperparameters that the authors report as best for long-form transcription, including beam search with 5 beams and temperature backoff. We also set a silence threshold of 20 seconds to skip long periods of silence.

274 trials had response times above 3 minutes due to errors. We truncated these trials' audio and marked them incorrect if they were above 181 seconds, allowing a second of lag.

Filtering irrelevant transcripts

Despite our best efforts to get participants to think aloud, some participants were either silent or did not say anything relevant to the task. We filter out these trials. We marked as irrelevant any trials where the transcript was a common string that Whisper outputs when tasked with transcribing silence, like "Thank you." When transcripts did not exactly match one of these strings, we used the LLM Llama 3.3 70B (Grattafiori et al., 2024) with a prompt to determine whether the transcript contained any content relevant to the Game of 24.

Coding transcripts as graphs

In analyzing the data, we assume that participants mentally search for a solution by working through a series of possible game states connected by arithmetic operations. Graphs are versatile data structures that are used extensively in search algorithms, so we use them to represent participants' search. Nodes in the graph represent game states—the sets of available numbers. Edges represent operations, which include combining any two numbers with an arithmetic operation or setting a subgoal. Subgoals are represented as backward edges from the goal state to the subgoal state. Figure 1b shows example graphs. We also kept track of the order in which people explored operations.

Graphs were generated by prompting a language model to write Python code using a custom "GraphBuilder" class with methods designed to help build a search graph easily. Using graph representations requires the general theoretical commitment that human reasoning in this task can be formulated as a search process, but we built only very general search operations into the class in order to avoid more specific theoretical commitments.

The language model had a system prompt containing instructions and the code for the custom class. We also included 10 in-context examples of transcripts with code translations written by the authors. Prompts also included the starting numbers for that trial, the response the participant submitted (if any), and the response time. We compared several LLMs for coding and used the results from the model that aligns best with human coding for subsequent analyses.

Automatic graph checking After getting a translation from a language model, we ran the code and automatically checked the graph it generates for common errors, like computing the resulting state incorrectly or using numbers that do not occur in the current state. When a graph had one of these problems, we called the language model again with a system prompt asking it to correct the code and in-context examples of code with errors and corrected code. We repeated this process up to five times, keeping the graph with the fewest problems. If a model failed to improve on a given iteration, we increased the temperature by 0.1 on the next iteration.

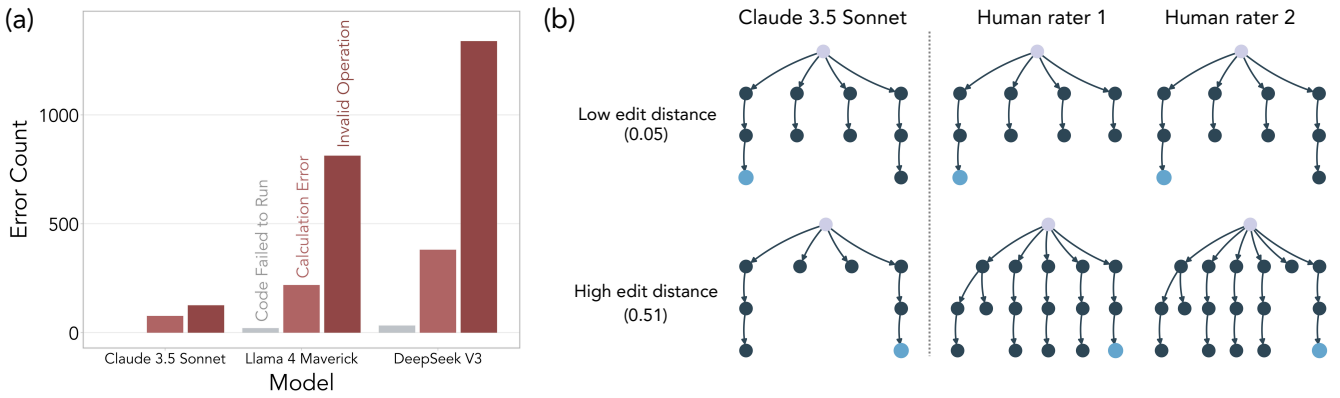


Figure 2: **Analysis of model-generated graphs**. (a) Counts of each type of automatically detected error for the three models that agree most with human coders. (b) Example graphs for trials with low and high mean human-model edit distance using Claude 3.5 Sonnet.

Results

Our analyses are organized around two goals: assessing the reliability and validity of the think-aloud method in our setting and examining characteristics of human reasoning traces.

Reliability and validity

Inter-rater reliability We evaluated seven language models for their ability to code human transcripts: Claude 3.5 Sonnet, a high-performing proprietary model, as well as six open-weights models: DeepSeek V3-0324, Llama 4 Maverick and Scout, Qwen 3 235B-A22B, Llama 3.3 70b, and Llama 3.1 8b (Anthropic, 2024; Liu et al., 2024; Team, 2025; Grattafiori et al., 2024). Performance was measured computing the average agreement with human coders. Two of the authors manually annotated a set of 50 randomly chosen trials, using the same graph checker and prompt as seen by LLMs, as well as a graph visualization function the LLMs did not have access to. We then computed the normalized graph edit distance between the graphs created by human and language model coders using the networkx Python package (Hagberg et al., 2008; Abu-Aisheh et al., 2015).[1]

As is shown in Figure 1c, agreement between human coders and language models is somewhat lower than inter-rater reliability between the human coders, with all language models exhibiting significantly less agreement with the humans than the humans do with each other ($p < 0.002$ in all cases, computed via permutation test). However, there is substantial overlap between the distributions of human-model and human-human edit distances, particularly for Claude 3.5 Sonnet, DeepSeek V3, and Llama 4 Maverick. Moreover, the model-produced graphs generally look sensible and qualitatively similar to human graphs, and capture important features like the path from the root node to the goal even at high disagreement (see Figure 2b). The human coders took on average 8 hours and 15 minutes to annotate 50 trials, which amounts to 816 hours to code all 4,947 trials at the same pace. In contrast, coding all trials with Claude took less than 2 hours and cost $74. While graphs produced by the models should be interpreted with some caution, they are close enough to human-generated graphs to be informative about human reasoning and enable analysis of reasoning traces at a scale that is not feasible using manual coding.

Error analysis We coded the full dataset with the three models that performed best in terms of graph edit distance: Claude 3.5 Sonnet, DeepSeek V3, and Llama 4 Maverick. After five iterations of automatically checking the graphs for errors and correcting them, some of the graphs still had errors. Figure 2a shows the total counts of each of the three types of errors that occurred: non-runnable code, incorrect resulting states, and operations that use numbers that do not exist in the states they start in. Claude produced the fewest errors, and never produced non-runnable code, so we use its annotations for our remaining analyses.

Effect of prompting to think aloud We compared participant performance between the think-aloud condition and the control condition to test whether thinking aloud affects performance. Participants in the control condition found the correct solution in 53% of trials, compared with 49% of trials on the same problems in the think-aloud condition. This difference came close to significance according to a permutation test ($p = 0.067$). A follow-up analysis found that participants in the think-aloud condition took significantly longer to respond on average (125.1 vs. 117.7 seconds, $p = 0.008$). The effect of thinking aloud on response times was similar across problems, as evidenced by a strong correlation between average problem-level response times across conditions ($r = 0.95$, $p < 0.0001$).

We also tested whether thinking aloud influences performance on some problems more than others—which would

[1] Graph edit distances are computed where nodes can only be matched to each other if they correspond to the same state. We normalize the graph edit distances by dividing them by the sum of the maximum vertex and edge counts. That is, $\max(|V_1|, |V_2|) + \max(|E_1|, |E_2|)$ Code that does not run is given an edit distance of 1.

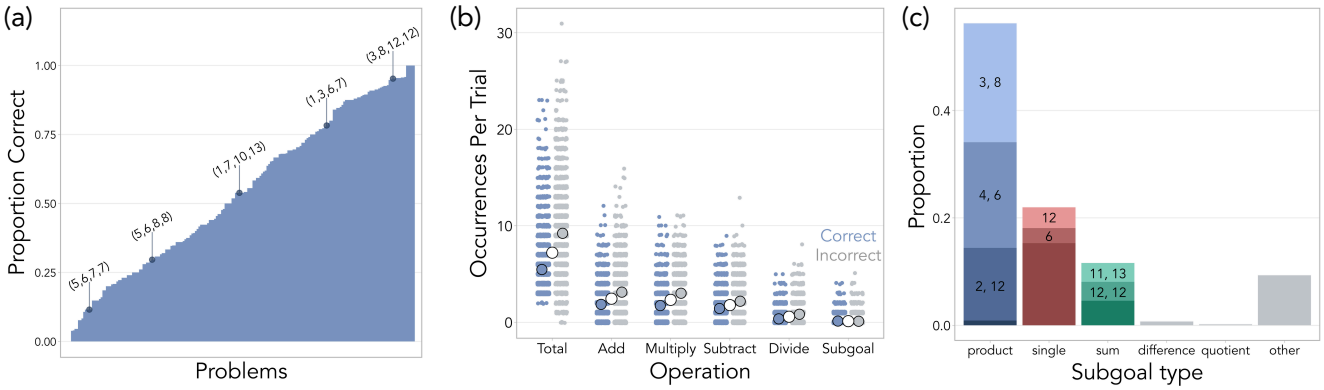


Figure 3: **Summary of participant data**. (a) Proportion of correct answers for every problem in the dataset. (b) Mean number of uses of each operation in trials where the participant was correct and incorrect. (c) Proportion of subgoals that are of each type, defined by how one can reach the goal from them.

suggest a change to the underlying reasoning processes. We computed split-half correlations for mean accuracy by problem in the think-aloud condition by randomly dividing the data into pairs of groups with 30 participants each (10 per stimulus group). We then compared the correlations between these groups to correlations between the think-aloud and control conditions, where participants were downsampled in the same way. We found no significant difference ($r = 0.83$ within think-aloud, $r = 0.79$ between conditions, $p = 0.74$).

Taken together, these results indicate that while asking participants to think aloud may slow them down, there is no evidence that it changes the pattern of which problems are easy and hard. This is consistent with findings from a prior meta-analysis (Fox et al., 2011). The large scale of our data enables us to robustly measure differences in overall performance and correlations between problem-level difficulties, which would not have been possible in a smaller study.

Characteristics of human reasoning traces

We next analyze aggregate patterns in how people search, including how often people use each type of operation, the extent to which they cluster around certain sequences of operations, and what leads to successful and unsuccessful search.

Use of operations A natural first question to ask is how often people consider each type of operation. Figure 3b shows the mean number of times each type of operation occurs in participants' search graphs, for trials where the participant got the answer correct and incorrect. Addition is the most common operation, followed closely by multiplication and subtraction, with division a distant fourth. This may indicate that in our task division is less cognitively natural to participants than other arithmetic operations. Participants explored more operations in incorrect trials, generating significantly more edges of all types except for subgoals. In successful trials, participants were able to efficiently reach the goal while considering relatively few operations. When participants did not get the right answer, they continued to try more operations.

Setting subgoals was rare: only 445 trials (9.0%) contain one or more subgoals and 58% of participants set no subgoals at all. Figure 3c shows the frequencies of the different types of subgoals, along with some of the most common individual states. Product subgoals, consisting of two factors of 24, were the most common by far. Many of the subgoals were single-number states. While it is not possible to reach 24 from these states, they tend to occur when the participant tries to make a number without mentioning the other numbers they would need to get to the goal, like "Can I make 12?" 12% of the subgoals were sum subgoals and almost none were differences or quotients. While participants explored addition and multiplication about equally, they set far more product subgoals than sum subgoals. Setting a product subgoal may be easier because there are only four natural number-valued states that could lead to the goal by multiplying, while there are 24 that could lead to the goal by adding.

Consistency and variation in search patterns Next, we analyze the extent to which participants use similar operations, and sequences of operations, to each other using Gini indices. This analysis follows McCarthy et al. (2023). The Gini index is a measure of dispersion, where higher numbers indicate more clustering: a value of 1 would mean all subsequences were identical, while a value of 0 would mean all subsequences were unique.[2]

Figure 4a shows the Gini coefficient for subsequence lengths 1-5 in humans and an agent that explores random legal actions matched to the humans in number of operations explored. Human search clusters much more around particular subsequences than the random agent, particularly for subsequences of lengths 2 and 3. This indicates that humans exhibit consistent multi-step reasoning strategies that extend beyond single operations.

[2]Due to the size of the search space, Gini indices were computed over only subsequences that occurred at least once in the data.

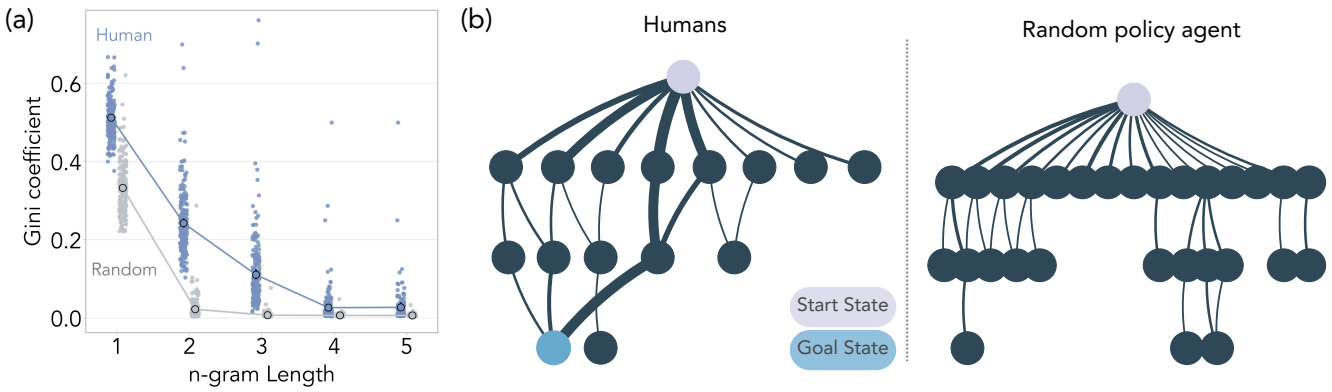


Figure 4: (a) Gini indices for sequences of operations of lengths 1 through 5 in humans (blue) and random search agents (gray). Each dot represents one Game of 24 problem. (b) Aggregate visualizations of human and random reasoning traces for one problem. Edge widths represent the number of times an operation was explored. Operations only explored once are filtered.

Predictors of success and failure in reasoning We can learn about people's reasoning strategies from when and how they succeed and fail at the task. Participants struggled with problems requiring division. The 29 problems in the experiment that could only be solved using division had a success rate of 20%, much lower than the 59% success rate for problems that can be solved without division ($p < 0.0001$). 47% of participants who got those problems wrong never tried division, suggesting that failures of consideration are a main driver of participants' reasoning failures.

Despite subgoals being rare overall, they appear to be helpful. Participants were significantly more likely to find the correct answer in trials where they set one subgoal than trials where they set none (65% vs. 53%, $p < 0.0001$). However, they were no more likely to find the correct answer when they set more than one subgoal than when they set none (47% vs. 53%, $p = 0.35$). This result suggests that working backward can help, even though people rarely do it. Setting multiple subgoals might indicate that a participant struggled with a problem and failed to achieve their first subgoal.[3]

Discussion

Reasoning is a hallmark of the human mind, but it is difficult to study from performance alone. The think-aloud method is a classic approach to eliciting information about the process of reasoning, but it has historically been limited by the amount of manual effort involved in coding. In this paper, we developed methods to automate think-aloud analysis in order to study human reasoning with both granularity and scale. We ran a large-scale experiment with 700 participants playing the Game of 24 and analyzed the data using an automated pipeline. We validated our methods by comparing them to human annotations. While the model reliability was lower than humans, it was high enough to be a favorable tradeoff given the substantial time and cost savings. From the graph representations, we learned that participants cluster around similar parts of the search space and discovered that people use addition and multiplication most often. Failing to consider division seemed to be a substantial impediment to participants' success. Our results illustrate the utility of think-aloud studies in gathering process-level data about reasoning.

Our automated pipeline overcomes the issue of scale, but our study still inherits some limitations of traditional think-aloud experiments. For instance, the verbal reports may not include everything a participant considers (Wilson, 1994; Ericsson & Simon, 1993), and our analysis pipeline may not have captured all of what they said. Secondly, our graph representations cannot express all of the participants' thoughts. For instance, "There has to be some multiplication here," does not fit into any of our graph operations. We used simple representations to study the operations people use, but different questions might require different representations. One potential solution to this is to employ more open-ended code generation from LLMs, as was done in a recent study in which an LLM generated code from participants' retrospective descriptions of their strategy in a sorting task (Xie et al., 2024b).

Automating think-aloud analysis enables us to overcome major limitations of traditional think-aloud studies. After an initial investment in the coding scheme and manually annotated examples, we were able to automatically code thousands of trials. This would simply not be feasible with manual coding. This automated method is also more reproducible than manual coding, as sharing prompts and in-context examples enables other researchers to obtain the same results. When using a closed-weights model, reproducibility is contingent on the model remaining available through an API. However, as open-weights models continue to improve, we expect reliable fully-open coding pipelines to be possible soon.

We hope that this work serves as a proof of concept that inspires the future development of automated methods for analyzing think-aloud data. We see the rich, granular data enabled by the think-aloud method at scale as a promising avenue for gaining new insights into how humans reason.

[3]These patterns were either weaker or non-significant using DeepSeek V3 and Llama 4 Maverick, yet note these models show substantially higher error rates compared to Claude 3.5 Sonnet.

Acknowledgments

The authors thank the Stanford Computation and Cognition Lab for feedback on early versions of this work. Particular thanks to Linas Nasvytis, Allison Chen, Samah Abdelrahim, and Veronica Boyce for their helpful comments on a draft of this paper. KG was supported by an HAI-SAP Grant and an NSF Expeditions grant.

References

Abu-Aisheh, Z., Raveaux, R., Ramel, J.-Y., & Martineau, P. (2015). An exact graph edit distance algorithm for solving pattern recognition problems. In *4th international conference on pattern recognition applications and methods 2015*.

Anthropic. (2024). *Claude 3.5 sonnet model card addendum.* Retrieved from https://www-cdn.anthropic.com/fed9cc193a14b84131812372d8d5857f8f304c52/Model_Card_Claude_3_Addendum.pdf

Besta, M., Blach, N., Kubicek, A., Gerstenberger, R., Podstawski, M., Gianinazzi, L., ... others (2024). Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 38, pp. 17682–17690).

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... others (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877–1901.

Ericsson, K. A., & Simon, H. A. (1993). *Protocol analysis: Verbal reports as data*. The MIT Press. doi: 10.7551/mitpress/5657.001.0001

Fox, M. C., Ericsson, K. A., & Best, R. (2011). Do procedures for verbal reporting of thinking have to be reactive? a meta-analysis and recommendations for best reporting methods. *Psychological bulletin*, *137*(2), 316.

Gandhi, K., Lee, D. H. J., Grand, G., Liu, M., Cheng, W., Sharma, A., & Goodman, N. (2024). Stream of search (sos): Learning to search in language. In *First conference on language modeling*. Retrieved from https://openreview.net/forum?id=2cop2jmQVL

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., ... others (2024). The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Hagberg, A., Swart, P. J., & Schult, D. A. (2008). *Exploring network structure, dynamics, and function using networkx* (Tech. Rep.). Los Alamos National Laboratory (LANL), Los Alamos, NM (United States).

Hartshorne, J. K., Tenenbaum, J. B., & Pinker, S. (2018). A critical period for second language acquisition: Evidence from 2/3 million english speakers. *Cognition*, *177*, 263–277.

Hertzum, M. (2024). Concurrent or retrospective thinking aloud in usability tests? A meta-analytic review. *ACM Transactions on Computer-Human Interaction*.

Kaplan, C. A., & Simon, H. A. (1990). In search of insight. *Cognitive psychology*, *22*(3), 374–419.

Larkin, J., McDermott, J., Simon, D. P., & Simon, H. A. (1980). Expert and novice performance in solving physics problems. *Science*, *208*(4450), 1335–1342.

Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., ... others (2024). Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. MIT press.

McCarthy, W. P., Kirsh, D., & Fan, J. E. (2023). Consistency and variation in reasoning about physical assembly. *Cognitive Science*, *47*(12), e13397.

Mechera-Ostrovsky, T., Ungermann, P., & Donkin, C. (2024). Using machine learning to automate the collection, transcription, and analysis of verbal-report data.

Newell, A., & Simon, H. A. (1961). GPS, a program that simulates human thought. *Rand Corporation Santa Monica, CA*.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.

Ostrovsky, T., & Newell, B. R. (2024). Verbal reports as data revisited: Using natural language models to validate cognitive models. *Decision*.

Peterson, J. C., Bourgin, D. D., Agrawal, M., Reichman, D., & Griffiths, T. L. (2021). Using large-scale experiments and machine learning to discover theories of human decision-making. *Science*, *372*(6547), 1209–1214.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2023). Robust speech recognition via large-scale weak supervision. In *International conference on machine learning* (pp. 28492–28518).

Russo, J. E., Johnson, E. J., & Stephens, D. L. (1989). The validity of verbal protocols. *Memory & cognition*, *17*, 759–769.

Team, Q. (2025, April). *Qwen3: Think Deeper, Act Faster.* https://qwenlm.github.io/blog/qwen3/.

Tong, L., Yang, J., Han, X., & Velasquez, L. (2014). The card game 24 and its application to math education. *International Journal of Mathematical Education in Science and Technology*, *45*(4), 624–633.

van der Maas, H. L., & Nyamsuren, E. (2017). Cognitive analysis of educational games: The number game. *Topics in cognitive science*, *9*(2), 395–412.

Van de Wiel, M. W. (2017). Examining expertise using interviews and verbal protocols. *Frontline Learning Research*, *5*(3), 112–140.

Van Someren, M., Barnard, Y. F., & Sandberg, J. (1994). *The think aloud method: a practical approach to modelling cognitive processes*. Academic Press.

Waterman, D. A., & Newell, A. (1971). Protocol analysis as a task for artificial intelligence. *Artificial intelligence*, *2*(3-4), 285–318.

Wilson, T. D. (1994). The proper protocol: Validity and completeness of verbal reports. *Psychological Science*, *5*(5), 249–252.

Xie, H., Xiong, H., & Wilson, R. (2024a). Evaluating predictive performance and learning efficiency of large language models with think aloud in risky decision making. In *Cognitive computational neuroscience (CCN).*

Xie, H., Xiong, H., & Wilson, R. C. (2023). Text2decision: Decoding latent variables in risky decision making from think aloud text. In *NeurIPS 2023 AI for science workshop.*

Xie, H., Xiong, H.-D., & Wilson, R. (2024b). From strategic narratives to code-like cognitive models: An llm-based approach in a sorting task. In *First conference on language modeling.*

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., & Narasimhan, K. (2023). Tree of Thoughts: Deliberate problem solving with large language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), *Advances in neural information processing systems* (Vol. 36, pp. 11809–11822). Curran Associates, Inc.