
De(ep)Composition: Deep Research for Argumentation by Query Decomposition*

Justin Blumencranz Jessie Dong Yucheng Jiang Monica S. Lam
Stanford University
{jmb25, jessiedong, yuchengj, lam}@stanford.edu

Abstract

Deep research systems answer a question by searching the web over many steps and writing a report with citations. Leading systems such as OpenAI Deep Research and STORM are built to summarize the evidence on every side, yet many tasks call for an argument that commits to a position. De(ep)Composition is a deep research pipeline for argument generation. It decomposes a question into a graph of sub-questions, answers the narrowest of them with web search, and composes the answers upward into a report that argues for a single position. In a blind evaluation on ten contested questions, De(ep)Composition took a firm position in 9 of 10 reports, compared with 1 for OpenAI Deep Research and 2 for STORM. It also attached citations to a larger share of its claims (0.80 against 0.68 and 0.60) and received the highest score for logical coherence (7.7 out of 10 against 6.5 and 6.0). Compared with STORM, its reports use less than half as many hedging words (2.8 against 6.4 per 1,000 words). An automatic citation check, which identifies the correct source for 96% of claims when citations are known to be right, finds that the cited source is the best match for only 14% of claims in the evaluated reports, close to the 10% expected by chance.

1 Introduction

Deep research systems automate the process of researching a question. A language model issues web searches over many rounds, reads the retrieved pages, decides after each round whether more searching is needed, and finally writes a report in which each statement cites the page it came from. OpenAI Deep Research (OpenAI, 2025), released in February 2025, and STORM (Shao et al., 2024), introduced by the Stanford OVAL lab in 2024, are two of the most widely used systems of this kind.

Both systems are designed to produce surveys. Asked whether the Federal Reserve should keep interest rates high, they present the evidence on each side and leave the judgment to the reader. A survey is useful for preliminary research, but many professional tasks end in a decision. A lawyer drafting a brief argues for one side of a case, and an analyst advising on an investment recommends whether to buy or sell. In both settings the reader expects a clear position, a small number of claims that support it, and evidence behind each claim, which a balanced survey does not provide.

The difference between a survey and an argument also suggests a different way to organize research. A survey can grow from the bottom up, because each search can suggest the next and the report describes whatever the searches find. An argument, by contrast, has a structure that is known before any search begins. The main claim depends on a few supporting claims, and each supporting claim depends on evidence that can be retrieved. Research for an argument can therefore be planned in advance, by decomposing the question into sub-questions that are each narrow enough to answer by search and then combining the answers in the order the argument requires.

*Code and data: <https://github.com/jessiedong01/deep-research-for-argumentation>

De(ep)Composition implements this plan. Its name joins the two parts of the method, deep research and decomposition. Given a question, a language model first generates a directed acyclic graph (DAG) of sub-questions, in which every internal node carries instructions for combining the answers of its children. The leaves are then answered by retrieval-augmented generation, and each parent is composed from its children in topological order. When a composed answer is incomplete, the pipeline adds a sub-tree of new questions beneath it. Finally, the answered graph is turned into an outline and a report that argues for the answer at the root. Figure 1 shows the three phases.

The pipeline was compared with OpenAI Deep Research and STORM in two evaluations, six automated debates and ten reports scored blind by one rater. It received the highest scores for argument quality and took a firm stance far more often than either baseline. A further analysis of the retained reports, which does not rely on a rater, supports the stance result but qualifies two others. Its reports cite densely but draw on few sources, and their conclusions are often more qualified than their titles. An automatic test of citation attribution then shows that the citations in the evaluated reports rarely point to the source a claim came from. The cause lies in how citation numbers were handled as answers were combined, and it applies to any system that composes cited text in stages.

Section 3 defines the task and the requirement that citations survive composition. Section 4 describes the pipeline, Sections 5 and 6 present the original evaluation and its limits, and Section 7 analyzes the content of the reports. Section 8 presents the attribution test and traces the citation failure to its source, and Section 9 states what the results do and do not support.

2 Related Work

Deep research systems. OpenAI Deep Research (OpenAI, 2025) is an agent that plans searches, browses the web, and writes a cited report. It was trained with reinforcement learning on browsing tasks of the kind collected in BrowseComp (Wei et al., 2025), where each answer is a short fact that is difficult to find. Its training objective therefore rewards locating and citing information rather than structuring an argument, and its intermediate reasoning is held in a single context window with no explicit dependencies between sub-questions. STORM (Shao et al., 2024) writes Wikipedia-like articles. It first identifies perspectives on a topic, then simulates conversations in which writers holding those perspectives question an expert grounded in search results, and finally organizes the answers into an outline and an article. Because its questions arise from these conversations as the research proceeds, its process is bottom-up, and its output is deliberately balanced. Both systems were designed for summarization, and the comparison below treats them as the strongest available summary-oriented baselines rather than as argumentation systems.

Decomposition and structured reasoning. Breaking a problem into sub-problems before solving it is an established prompting strategy. Least-to-most prompting (Zhou et al., 2023), decomposed prompting (Khot et al., 2023), and self-ask (Press et al., 2023) each have a model generate sub-questions whose answers feed the original question. Tree of Thoughts (Yao et al., 2023a) searches over intermediate reasoning steps, ReAct (Yao et al., 2023b) interleaves reasoning with calls to external tools, and Graph of Thoughts (Besta et al., 2024) generalizes the tree to a graph in which intermediate results can merge. These methods target problems with verifiable answers, such as arithmetic, puzzles, and short-form question answering. They do not synthesize prose from retrieved evidence, and they do not revise their reasoning structure when a combined answer proves incomplete. De(ep)Composition adopts the decomposition idea and extends it with explicit composition instructions, leaves grounded in retrieved evidence, and refinement of the graph when gaps appear.

Multi-agent frameworks. AutoGen (Wu et al., 2023), CAMEL (Li et al., 2023), and natural-language societies of mind (Zhuge et al., 2023) coordinate several language-model agents through conversation, assigning roles such as planner, researcher, and critic. In these frameworks the agents exchange messages, but the structure of the task itself is not represented, so the output of one agent does not formally constrain how another combines

it. De(ep)Composition makes that structure explicit in the DAG, so that the answer of each sub-question constrains the composition of its parent.

Argument mining and generation. Argument mining identifies claims, premises, and the relations between them in existing text (Lawrence & Reed, 2020). Argument generation produces new arguments. Hua & Wang (2018) generate counter-arguments with a sequence-to-sequence model conditioned on retrieved evidence, and Project Debater (Slonim et al., 2021) assembles debate speeches from a large mined corpus of claims and evidence. These systems draw on fixed corpora and generate text in a single pass. De(ep)Composition instead retrieves evidence from the live web and constructs the structure of the argument before filling it in.

Citation and attribution. Text generated with citations can be checked for whether each citation supports the sentence it accompanies. Rashkin et al. (2023) formalize this requirement as attribution to identified sources. Liu et al. (2023) applied human judgments to four generative search engines and found that only about half of the generated sentences were fully supported by their citations, and that about three quarters of the citations supported their sentence. Gao et al. (2023) built a benchmark that measures citation precision and recall automatically with an entailment model. These studies evaluate generation from retrieved passages in a single step. Section 8 examines a multi-step setting, in which citations must remain correct through several rounds of composition, and uses a lexical test that requires no additional model.

Implementation. The pipeline is written with DSPy (Khatab et al., 2024), which expresses each language-model call as a signature, consisting of inputs, outputs, and an instruction, and allows these calls to be composed into a program. The signatures used here are listed in Appendix A.

3 Problem Formulation

Argumentative report generation. The input is a motion m , a statement such as “The U.S. Federal Reserve should prioritize maintaining high interest rates over stimulating growth.” The output is a report R that takes a position on m , supports it with claims, and attaches to every factual claim a citation to a retrieved source. A good report has three properties. It is *decisive*, meaning that it states one position and holds it throughout. It is *grounded*, meaning that each cited claim is supported by the source it cites, the property that attribution studies call verifiability (Rashkin et al., 2023, Liu et al., 2023). It is *connected*, meaning that its claims combine into a case for the position rather than forming a list of separate facts.

Research graphs. De(ep)Composition represents the research behind a report as a directed acyclic graph $G = (V, E)$ with a single root r whose question is m . Each node v carries a question q_v , an expected answer format f_v , an answer a_v , and a set of sources S_v . An edge (u, v) indicates that q_v is a sub-question of q_u . A leaf v is answered by retrieval, $a_v = \text{ANSWER}(q_v, S_v)$, where S_v is the set of documents found for q_v . A parent u with children $C(u)$ is answered by composition,

$$a_u = \text{COMPOSE}(q_u, \{a_v : v \in C(u)\}, \kappa_u), \quad S_u = \bigcup_{v \in C(u)} S_v, \quad (1)$$

where κ_u denotes the composition instructions that state how the answers of the children are to be combined. The report is then written from the answered graph, $R = \text{WRITE}(m, G)$.

Provenance. A report can be grounded only if its citations remain correct through every composition step. Let $\text{cite}(a_v)$ denote the set of (claim, source) pairs in a_v . Composition preserves provenance if every pair that a_u takes from a child is a pair that the child itself contains. Under this condition, any citation in the report can be traced through the graph to the leaf at which its source was read. Section 8 tests whether the pipeline preserves provenance and identifies where it does not.

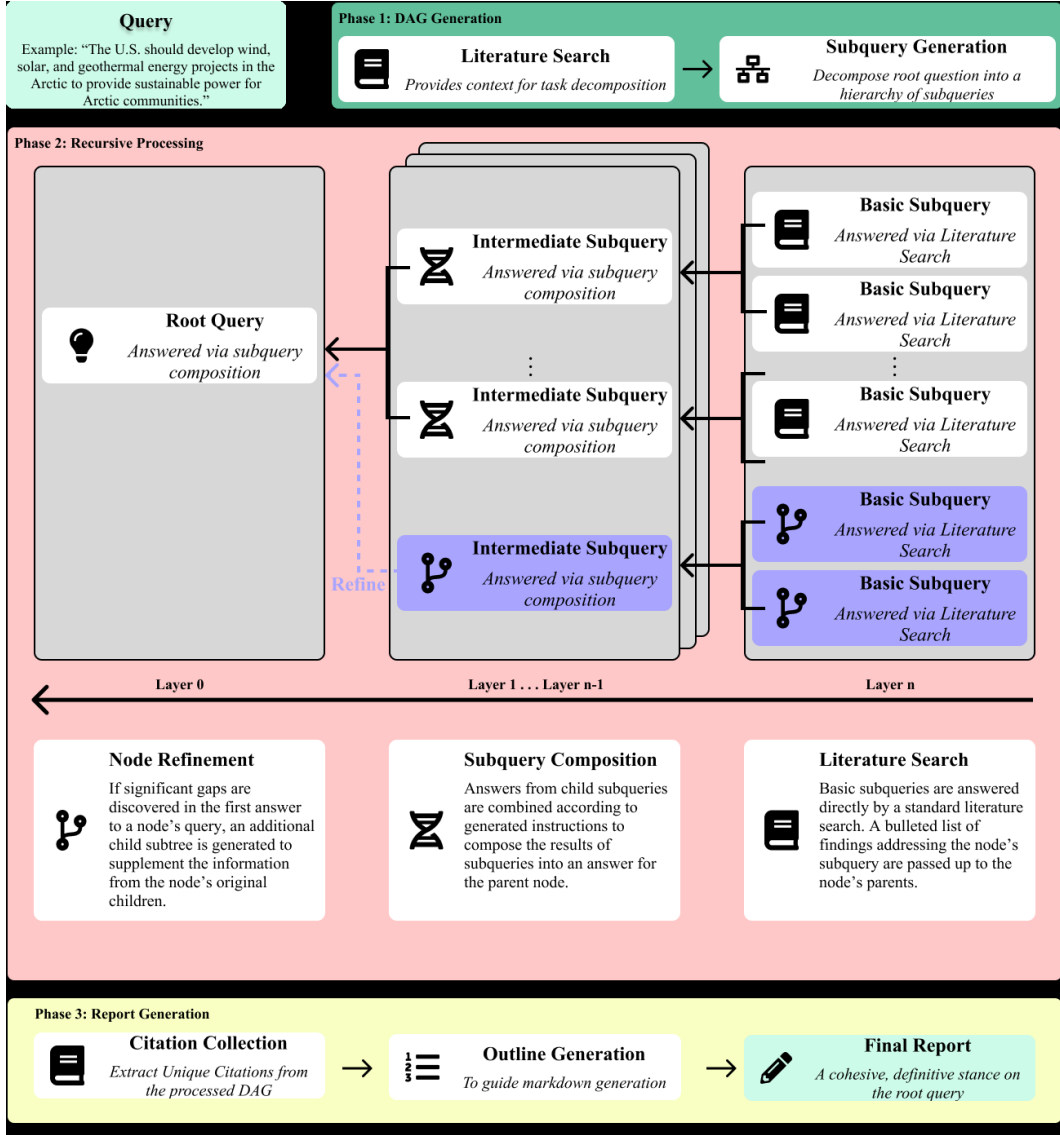


Figure 1: The three phases of De(ep)Composition. In Phase 1, a literature search provides context for decomposing the question into a DAG of sub-questions. In Phase 2, leaves are answered by literature search, parents are composed from their children, and nodes whose answers are incomplete are refined with new sub-trees. In Phase 3, the answered DAG is turned into an outline and a report that argues for the root answer.

4 Method

De(ep)Composition takes a research question and returns a report that argues for an answer to it, with inline citations to web sources. It runs in three phases, namely DAG generation, recursive processing, and report generation. This section describes the pipeline as it was when the reports in Section 5 were generated, and Appendix C describes the code versions involved. Every language-model call is implemented as a DSPy signature (Khatab et al., 2024), and the signatures are reproduced in Appendix A.

Literature search. The planning phase and the leaf nodes share one literature-search routine. A planning model turns a question into web queries. A retriever fetches

the results through the Serper API, scrapes the pages, embeds their passages with text-embedding-3-large, and keeps the passages closest to the query. An answer model then writes a cited answer from these passages. After each round, a completeness check decides whether another search is needed, up to a fixed number of retriever calls per question.

Phase 1, DAG generation. A short literature search on the root question first gives the planner context about the topic. The planner then builds the DAG one layer at a time. For each node in the current layer it generates child sub-questions, conditioned on the node’s question, the literature summary, the children generated so far, and the structure of the whole graph. Conditioning on the whole graph serves two purposes. It discourages siblings from overlapping, and it encourages the children of each node to cover its question jointly. The depth of the graph, the total number of nodes, and the number of children per node are capped. A second pass then writes composition instructions for every parent. Given the parent’s question, its expected answer format, and the questions of its children, the model states how the children’s answers should be combined into the parent’s answer. These instructions refer to children by their identifiers, which makes each dependency explicit.

Phase 2, recursive processing. The DAG is sorted topologically and processed from the leaves to the root, since a parent can only be composed once its children have been answered. Leaves are answered in parallel by literature search, and the answer model writes bullet points in the node’s expected format while keeping every citation from the search. Parents are then composed in sequence. The children’s answers, the composition instructions, and the expected format are passed to a synthesis model, which may use only the content of the children, must cite using the children’s citation numbers, and must state explicitly when the children contradict each other. Because a parent draws only on its children, every claim in its answer should trace back to evidence retrieved at a leaf.

A plan made before research can still omit something the research later shows to matter. After composing a parent, a gap explorer therefore lists possible omissions in the composed answer, and a gap evaluator decides, at a configurable level of conservatism, whether any omission is important enough to act on. If one is, the pipeline generates a sub-tree of new sub-questions that address the omission, attaches it beneath the parent, processes it in the same way as the rest of the graph, and composes the parent again. Each parent is refined at most once in the configuration used here.

Phase 3, report generation. The answered DAG is serialized as text, and its citations are collected into a single numbered list. An outline model, instructed to argue for a position on the root question, writes an outline whose sections follow the branches of the graph. A report model then writes the report from the outline and the graph. It keeps the citations, adopts the position of the root answer, and titles the report with the claim rather than the question. Finally, citation numbers are normalized and a bibliography is appended.

Cost. Let $|V_{\text{leaf}}|$ and $|V_{\text{par}}|$ denote the numbers of leaves and parents after refinement. Each leaf requires one literature search, which comprises at least a query-generation call, a retrieval step with embedding, an answer call, and a completeness check. Each parent requires one composition call per round and one gap check. Planning adds one call per parent for its children and one for its instructions, and writing adds two calls. The number of model calls is therefore approximately $4|V_{\text{leaf}}| + (3 + 2\rho)|V_{\text{par}}| + 3$. This number grows linearly with the size of the graph, and the leaves can be processed in parallel. The logged run in Appendix B has 6 leaves and 4 parents and so makes on the order of 40 model calls, compared with one or a few calls for a report written in a single pass.

Models and limits. The evaluation runs used GPT-4.1-mini for answer generation during retrieval and for planning, GPT-5 for DAG generation, synthesis, and report writing, and text-embedding-3-large for retrieval, all accessed through Azure OpenAI. The default limits cap the graph at depth 2, 50 nodes, and 10 children per node, with one retriever call per literature search and one refinement per parent.

Algorithm 1 De(ep)Composition

Require: motion m , limits on depth d , nodes n , and children c , refinement rounds ρ

```
1:  $L \leftarrow \text{LITERATURESEARCH}(m)$  ▷ context for planning
2:  $G \leftarrow \text{root } r \text{ with } q_r = m$ 
3: for each layer  $\ell = 0, \dots, d - 1$  do ▷ Phase 1
4:   for each node  $u$  at depth  $\ell$  do
5:     add children  $\text{PLANCHILDREN}(q_u, L, G, c)$  under  $u$ , within the node limit  $n$ 
6:   end for
7: end for
8: for each parent  $u$  do  $\kappa_u \leftarrow \text{COMPOSITIONINSTRUCTIONS}(q_u, \{q_v : v \in C(u)\})$ 
9: end for
10: for each node  $u$  in topological order, leaves first do ▷ Phase 2
11:   if  $u$  is a leaf then
12:      $(a_u, S_u) \leftarrow \text{ANSWERFROMSEARCH}(q_u)$ 
13:   else
14:     for  $k = 0, \dots, \rho$  do
15:        $a_u \leftarrow \text{COMPOSE}(q_u, \{a_v\}_{v \in C(u)}, \kappa_u)$ ,  $S_u \leftarrow \bigcup_{v \in C(u)} S_v$ 
16:        $\Delta \leftarrow \text{FINDGAPS}(q_u, a_u)$ 
17:       if  $k = \rho$  or  $\Delta = \emptyset$  then break
18:     end if
19:     attach a sub-tree for  $\Delta$  under  $u$  and process it as in Phase 2
20:   end for
21: end if
22: end for
23:  $O \leftarrow \text{OUTLINE}(m, a_r, G)$ ,  $R \leftarrow \text{REPORT}(O, m, a_r, G)$  ▷ Phase 3
24: return  $R$  with a bibliography of  $\bigcup_v S_v$ 
```

5 Experiments

Two evaluations compare De(ep)Composition with OpenAI Deep Research (OpenAI, 2025) and STORM (Shao et al., 2024). The first is interactive and consists of automated debates between De(ep)Composition and Deep Research, scored by a language model. The second is a blind human assessment of full reports written by all three systems. Both evaluations are small, and their size is stated here and discussed further in Section 9.

Automated debates. Three motions were chosen to require different kinds of reasoning. The first is moral and states that moral systems rooted in theism are preferable to non-theistic ones. The second is economic and technical and states that U.S. semiconductor manufacturers will outperform their global peers over the next decade because of the CHIPS and Science Act. The third is conceptual and states that AI systems can be creative in the way humans are. Each motion was debated twice, with De(ep)Composition and Deep Research exchanging the Pro and Con sides, which yields six transcripts and controls for any advantage of one side over the other. Each system researched the motion in its own way. De(ep)Composition decomposed the motion into a DAG and built its case from the answered nodes, while Deep Research ran its default retrieval and synthesis process. The exchange itself was generated by GPT-5-mini, which wrote alternating turns of 200 to 300 words for each side from that side’s research and the opponent’s previous turn. GPT-4o-mini then scored each transcript on stance consistency, defined as the fraction of turns that hold the assigned side, and on logical coherence, evidence integration, rebuttal specificity, persuasiveness, and fluency, each on a scale from 1 to 10. The scoring followed the framework of DeepResearch Bench, an unpublished course project by Abhinav Chinta and Charles Ding.

Human assessment of reports. Ten motions were selected so that each requires evidence synthesis and a position rather than factual recall, and together they span ethics, economics, policy, and technology. Table 7 lists them. Each system wrote one full report per motion. One author scored the thirty reports with the identity of the system hidden. The rubric

Metric	De(ep)Composition	OpenAI Deep Research	Difference
Stance consistency (%)	91	68	+23
Logical coherence (1–10)	8.8	7.2	+1.6
Evidence integration (1–10)	8.9	7.1	+1.8
Rebuttal specificity (1–10)	8.7	7.0	+1.7
Persuasiveness (1–10)	8.5	7.6	+0.9
Fluency (1–10)	7.7	8.6	−0.9

Table 1: Automated debate scores, averaged over six transcripts (three motions, both sides), judged by GPT-4o-mini.

Metric	De(ep)Composition	OpenAI Deep Research	STORM
Supported-claims rate	0.80	0.68	0.60
Evidence utilization	0.76	0.67	0.61
Logical connection quality (1–10)	7.7	6.5	6.0
Firm stance (of 10 reports)	9	1	2

Table 2: Human-coded scores on ten reports per system, one blind rater, means over reports.

measured the supported-claims rate, defined as the fraction of claims that carry a citation, the evidence utilization, defined as the fraction of cited sources that the argument actually uses, the quality of logical connections on a scale from 1 to 10, and whether the report holds a firm position throughout. The rubric also included evidence relevance and originality, but the recorded scores for these two measures were identical across the three systems and could not be reconstructed, so they are omitted. Because there is a single rater, no agreement statistic can be computed, and the values reported below are means over ten reports without confidence intervals.

Retained materials. The De(ep)Composition and STORM reports for the ten motions are available in the repository, together with the prompts, the pipeline, and one complete logged run. The Deep Research reports and the debate transcripts were not retained. Sections 7 and 8 therefore analyze the twenty retained reports and the logged run.

6 Results

Automated debates. Table 1 reports the judge’s scores. De(ep)Composition scored higher than Deep Research on stance consistency, logical coherence, evidence integration, rebuttal specificity, and persuasiveness, and lower on fluency. The transcripts suggest a common explanation across the three motions. In the theism debate, De(ep)Composition’s Pro case began from the claim that moral objectivity requires a transcendent foundation and supported it with sub-claims that each carried evidence, such as rates of volunteering in religious populations. Deep Research’s Con case relied on the Euthyphro dilemma and repeated it across turns with little new evidence. With the sides reversed, De(ep)Composition argued the Con case through meta-ethical consistency, cross-cultural variation in moral codes, and secular moral psychology, while Deep Research’s Pro turns restated earlier points. In the semiconductor debate, De(ep)Composition divided its Pro case into measurable claims about the cost of capital under the subsidies, the integration of research with fabrication, the resilience of supply chains, and projected yields. Its Con case rested on quantified dependencies, including a projected plateau in the U.S. share of fabrication capacity near 14% by 2032. Deep Research’s cases on both sides remained qualitative. In the creativity debate, De(ep)Composition separated an empirical branch, which cited benchmarks of divergent thinking, from a conceptual branch, which defined creativity as the combination of novelty and usefulness. Deep Research argued from metaphysical objections without an operational definition. Across the six transcripts, De(ep)Composition made roughly twice as many claims linked to evidence. Each of these differences follows

Measure	De(ep)Composition	STORM	Difference	95% CI	Wins
Citations per 100 words	5.7	4.7	+1.0	[+0.0, +2.1]	8/10
Sentences with a citation	0.46	0.60	-0.15	[-0.23, -0.09]	0/10
Distinct sources	8	27	-18	[-24, -14]	0/10
Citations per source	9.3	4.1	+5.0	[+1.7, +7.2]	10/10
Hedges per 1,000 words	2.8	6.4	-2.2	[-7.3, -0.4]	1/10
Contrast markers per 1,000 words	5.5	6.0	-0.3	[-0.8, +1.0]	4/10
Commitment markers per 1,000 words	6.8	3.8	+3.3	[+0.4, +4.2]	10/10
Causal connectives per 1,000 words	4.2	3.0	+0.3	[-0.5, +1.8]	5/10
Words	1339	2426	-1172	[-1621, -617]	0/10

Table 3: Paired comparison of the ten De(ep)Composition and ten STORM reports. Values are medians over motions. The difference is the median of the per-motion differences, with a 95% bootstrap interval, and “Wins” counts the motions on which De(ep)Composition is higher. Rates use the word lists in Appendix F.

from the structure of the pipeline, which forces every sub-claim to be answered and supported before the case is assembled. The one measure on which Deep Research scored higher is fluency. Its turns read more smoothly, whereas De(ep)Composition’s dense citation of sub-claims interrupts the flow of the prose.

Human assessment of reports. Table 2 reports the rater’s scores. De(ep)Composition had the highest supported-claims rate (0.80 against 0.68 for Deep Research and 0.60 for STORM), used the largest share of its cited sources (0.76 against 0.67 and 0.61), and received the highest score for logical connection (7.7 against 6.5 and 6.0). It held a firm position in 9 of 10 reports, whereas Deep Research did so in 1 and STORM in 2. The rater’s notes describe how these differences appear in the text. On the motion concerning a moral duty to strangers, De(ep)Composition built a layered argument that linked empathy to social trust, Deep Research gave a balanced account that reached no decision, and STORM described the positions without connecting them causally. On the Federal Reserve motion, De(ep)Composition argued through a chain from inflation control to wage stability, while Deep Research summarized both effects without relating them. On the motion concerning investment in foundation models, De(ep)Composition produced a conditional argument, holding that foundation models concentrate capital efficiency unless saturation reduces marginal returns, which neither baseline matched.

Two of these differences have plausible sources in the design of the pipeline. The higher supported-claims rate is expected because every leaf answer is written from search results with citation markers and every parent is composed only from its children, which leaves an uncited claim little route into the report. This measure, however, records whether a claim carries a citation and not whether the citation is correct, a question that Section 8 addresses directly. The higher rate of firm positions follows in part from the report prompt, which instructs the model to argue for the root answer, and Section 9 returns to this point.

7 Analysis of the Reports

The scores in Table 2 come from a single rater, so they are vulnerable to that rater’s judgment. The twenty retained reports, ten from De(ep)Composition and ten from STORM on the same motions, allow a second assessment that does not depend on any rater. Each report is measured in the same way, and the two systems are compared motion by motion, which controls for differences in difficulty between motions. Because the Deep Research reports were not retained, this section compares De(ep)Composition with STORM only. All differences are medians of the paired per-motion differences, reported with 95% bootstrap intervals over motions (Efron & Tibshirani, 1993) and Wilcoxon signed-rank tests (Wilcoxon, 1945). With ten pairs, the smallest attainable two-sided p -value is 0.002. Bibliographies, headings, and markup are removed before counting.

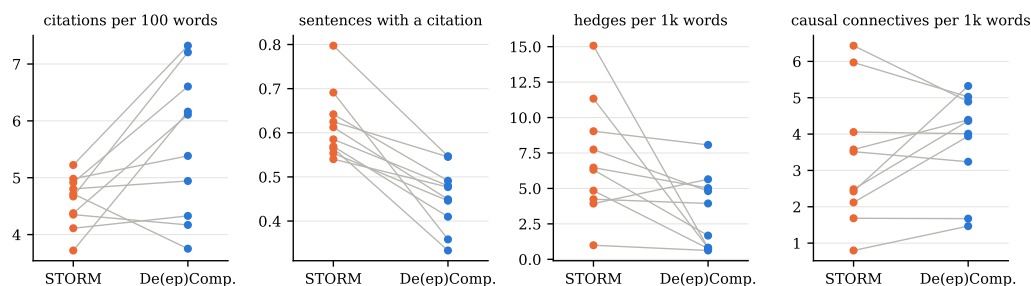


Figure 2: Four measures per motion. Each line joins the STORM report (left) and the De(ep)Composition report (right) for one motion.

Stance. If the rater’s judgment of stance is correct, it should be visible in the language of the reports. Hedges are words such as *may*, *could*, and *suggests* that withhold commitment (Hyland, 2005), and commitment markers are words such as *must*, *demonstrates*, and *therefore* that assert it. De(ep)Composition’s reports use fewer hedges than STORM’s, 2.8 per 1,000 words against 6.4, and the difference holds on nine of ten motions ($p = 0.020$). They also use more commitment markers, 6.8 per 1,000 words against 3.8, on all ten motions ($p = 0.002$). Contrast markers such as *however* and *on the other hand* occur at similar rates in both systems (5.5 and 6.0, $p = 0.922$). De(ep)Composition therefore acknowledges opposing views about as often as STORM does. It differs in committing to a position after doing so.

The commitment is strongest at the beginning of a report and weaker at its end. Table 5 records, for each De(ep)Composition report, the side of the motion it takes, whether its title states a claim, and how its final paragraph concludes. Of the ten titles, 9 state a claim, as the report prompt requires, whereas every STORM report opens with a lead section that frames the question as a debate. The final paragraphs are less committed. Of the ten, 5 end firmly, 2 qualify the claim of the title, 2 make it conditional, and 1 replaces a yes-or-no answer with a compromise. For example, the creativity report concludes that human creativity remains irreducible “but only provisionally,” and the theism report concludes that the preference for theistic morality “depends fundamentally on one’s valuation” of transcendent truth. One report also argues for the side opposite to the motion as worded, holding that application-layer startups will outperform developers of foundation models. A plausible explanation is that the stance instruction controls the title and executive summary, which are written first, whereas the conclusion is written last from a graph whose branches cover both sides of the question, so the balanced material reappears at the end.

Connection. The rater scored De(ep)Composition higher on logical connection, and the original assessment attributed this to causal chains between sub-claims. Causal connectives such as *because*, *leads to*, and *as a result* provide a rough lexical check of this explanation. Their rate does not differ reliably between the systems, at 4.2 per 1,000 words against 3.0, with De(ep)Composition higher on five of ten motions ($p = 0.432$). A count of connectives captures only part of what makes an argument coherent, so this result does not contradict the rater, but it provides no independent support for the explanation either.

Length and sourcing. De(ep)Composition’s reports are shorter than STORM’s on every motion, with a median of 1339 words of body text against 2426. They cite more densely, at 5.7 citations per 100 words against 4.7 ($p = 0.037$), but they draw on far fewer sources. The body of a De(ep)Composition report cites a median of 8 distinct sources, compared with 27 for STORM, and each source is cited 9.3 times on average against 4.1. Despite this density, a smaller share of De(ep)Composition’s sentences carry any citation (0.46 against 0.60, lower on every motion). The two observations are consistent because De(ep)Composition concentrates its citations in bullet points of evidence and leaves the connecting prose

Source type	Sources	Share
Government	45	13%
Academic	74	21%
Think tank or NGO	78	22%
News or magazine	3	1%
Encyclopedia	4	1%
Company, blog, or other	145	42%

Table 4: Domains of the 349 sources listed in De(ep)Composition’s ten bibliographies, classified by host name.

uncited. The bibliographies show the same concentration from the other direction. Each De(ep)Composition report lists a median of 36 retrieved sources (range 30 to 39) and separates them into cited and uncited sources, but the body cites only 7 to 13 of them, a median of 25%. Most of the evidence that the pipeline retrieves never reaches the report.

Sources. Table 4 classifies the 349 sources in De(ep)Composition’s bibliographies by domain. Government sites account for 13% of them, academic publishers and university sites for 21%, think tanks and other non-profit organizations for 22%, and company sites, blogs, and other pages for the largest share, 42%. News sites are almost absent. Of the 349 links, 69% still resolved when checked in October 2026, eleven months after the reports were written, 27% refused automated requests, and 4% no longer existed or returned errors. The STORM exports contain citation numbers but no URLs, so the same check cannot be applied to STORM.

Structure. The two systems organize their reports in different ways. STORM writes in the form of an encyclopedia article, with a lead section that frames the question, a background section, and sections for the arguments on each side, and it often closes on the opposing case. De(ep)Composition writes in the form of a brief, with a claim as the title, an executive summary that states the answer, sections that follow the branches of the graph, and a conclusion. On the motion concerning the African Continental Free Trade Area, STORM opens by describing the agreement and the history of its ratification, while De(ep)Composition opens by stating that the evidence “overwhelmingly supports” the conclusion that the benefits outweigh the harms. On the motion concerning a moral duty to strangers, STORM ends with critics who find an equal obligation to strangers unrealistic, and De(ep)Composition ends with a synthesis in which “impartiality and partiality are not opposites,” a softer position than its title asserts.

8 Citation Fidelity

The rubric in Section 5 counted a claim as supported when it carried a citation. It did not check whether the cited source actually states what the claim says. Studies of attribution in generative search show that this second check matters, since Liu et al. (2023) found that about a quarter of citations did not support their sentence. This section measures citation fidelity for De(ep)Composition. It introduces an automatic test, validates it on citations known to be correct, applies it to the evaluated reports, and identifies the step in the pipeline at which citations lose their link to their sources.

8.1 A lexical attribution test

The test rests on a simple observation. A sentence written from a source tends to share distinctive words with that source, so the source it came from should be more similar to it than most other sources are. For every sentence with bracketed citations, the test therefore ranks all sources available to the document by TF-IDF cosine similarity (Salton & Buckley, 1988) between the sentence and the source text. It then records the rank of each cited source as a percentile, from 0 for the best match to 1 for the worst. If citations point to the sources

Motion	Side	Title	Final paragraph
Human creativity will not remain fundamentally irreducible to...	for	claim	qualified
Individuals have a moral duty to prioritize the well-being of...	for	claim	qualified
Investing in companies developing foundation models -e.g., OpenAI,...	against	claim	firm
Israel and Palestine should pursue a long-term joint economic zone...	for	claim	firm
It is ethically justified for hedge funds to use alternative data...	for	topic	conditional
Moral systems rooted in theism are preferable to non-theistic...	for	claim	conditional
The benefits of the African Continental Free Trade Area outweigh...	for	claim	firm
The U.S. Federal Reserve should prioritize maintaining high...	neither	claim	compromise
The U.S. should develop wind, solar, and geothermal energy...	for	claim	firm
U.S. semiconductor manufacturers will outperform global peers over...	for	claim	firm

Table 5: Position of each De(ep)Composition report. The side is given relative to the motion as worded. The final paragraph is *firm* if it restates the title’s claim, *qualified* if it limits the claim, *conditional* if it makes the claim depend on unmet conditions or values, and *compromise* if it answers neither side.

from which claims were written, cited sources should rank near the top. If citation numbers are effectively random, their percentiles should be spread evenly between 0 and 1, with a median of 0.5. Two statistics summarize the result, namely the median percentile of the cited sources and the share of claims whose single best-matching source is among those cited, which is compared with the share expected by chance. The test requires no model calls and can be applied to every report. Because it is lexical, a source that expresses a claim in different words can rank lower than it should, and the validation below measures how much this limitation matters in practice.

Validation. A test of this kind is useful only if it can recognize correct citations. The leaf answers of the logged run (Appendix B) provide a set of citations that are correct by construction, because each answer is written in a single model call directly from the retrieved excerpts it cites. Against all 26 sources of the run, the cited source is the best match for 0.96 of leaf claims, compared with 0.10 expected by chance, and the median percentile of the cited sources is 0.04 (95% interval 0.00 to 0.08). The evaluated reports cite full web pages rather than excerpts, which contain more unrelated text. The test was therefore repeated using the full text of the logged run’s pages, fetched for this analysis. On the 10 pages that returned usable text, the cited page is the best match for 0.84 of claims, compared with 0.13 by chance. On both kinds of text, the test clearly separates correct citations from random ones.

8.2 Results

Table 6 and Figure 3 present the results. The ten evaluated reports contain 235 cited claims and 491 pairs of a claim and a cited source. Among all sources listed in each report, the median percentile of the cited sources is 0.47 (0.40 to 0.53), which cannot be distinguished from the value of 0.5 expected under random citation. The best-matching source is cited for 0.14 of claims, close to the 0.10 expected by chance. Restricting the candidates to the sources that a report actually cites, a median of 6 per report, gives the same picture, with the best match cited for 0.41 of claims against 0.36 by chance. The final report of the logged run, produced by the same report-writing step, also performs at chance, with a median

Setting	Claim–source pairs	Median rank percentile	Top-3 hit rate	Chance	Best match cited
Leaf answers, own numbering	63	0.04 [0.00, 0.08]	0.83	0.12	0.96 (chance 0.10)
Leaf answers, own numbering, full pages	24	0.00 [0.00, 0.11]	0.88	0.30	0.84 (chance 0.13)
Leaf answers as a parent reads them, global numbering	63	0.04 [0.00, 0.08]	0.83	0.12	0.96 (chance 0.10)
Leaf answers as a parent read them, original pipeline	63	0.12 [0.08, 0.16]	0.49	0.12	0.52 (chance 0.10)
Final report, logged run	173	0.48 [0.44, 0.52]	0.06	0.12	0.04 (chance 0.10)
Evaluated reports, all listed sources	491	0.47 [0.40, 0.53]	0.18	0.14	0.14 (chance 0.10)
Evaluated reports, cited sources only	491	0.43 [0.40, 0.50]	0.56	0.51	0.41 (chance 0.36)

Table 6: Attribution of cited claims. Each row ranks every available source by lexical similarity to each cited claim. The median rank percentile of the cited sources is 0 if they are always the best match and 0.5 if citations are random (95% bootstrap interval in brackets). The top-3 hit rate is the share of cited sources ranked in the top three, and the chance column gives its expected value under random citation. The last column is the share of claims whose best match is cited.

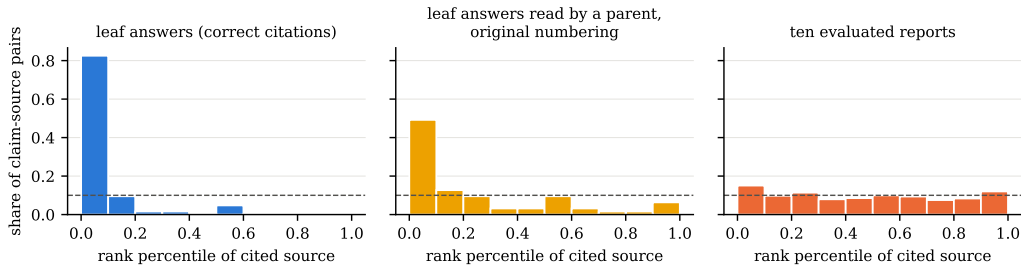


Figure 3: Rank percentile of each cited source among all available sources. Correct citations pile up at 0 (left). The evaluated reports are flat, as random citations would be (right). The middle panel shows the leaf answers as they are read by a parent node during composition (Section 8.3). The dashed line marks the uniform distribution.

percentile of 0.48. By this test, the citations in the evaluated reports carry little information about which source supports which claim.

Two examples from the theism report illustrate the result. A bullet stating that “non-theistic systems such as Confucianism, Buddhism, and Stoicism achieved moral stability” without divine law cites five sources, all of which argue for the existence of God from the moral law. A bullet on ancient Egyptian, Israelite, and Vedic societies cites “A Beginner’s Guide to the Moral Argument.” The claims may well be accurate, but the cited sources are not where they came from.

8.3 Mechanism

The code that produced the evaluated reports is not available (Appendix C). The repository version from a week earlier is available, however, together with a run it produced, and it loses the link between claims and sources at two steps. Both steps are consistent with the pattern observed in the evaluated reports, although they cannot be confirmed in the code of the evaluated version itself.

Local numbers read against a merged list. Each leaf answer numbers its own sources from [1] to [k]. When a parent is composed, the pipeline merges the sources of its children into a single deduplicated list numbered from [1] to [m] and instructs the synthesis model to cite by the numbers in that list. The children’s answers, however, are passed with their original local numbers. A citation “[1]” copied from the second child therefore refers, in the merged list, to the first source of the first child. The middle rows of Table 6 reproduce this step on the logged run. When the leaf claims’ citations are read against the parent’s merged list, the best match is cited for only 0.52 of claims, mostly those taken from each parent’s first child, compared with 0.96 in the leaves’ own numbering. Because every level

of the graph repeats the merge, the misattribution compounds as answers move toward the root.

Node identifiers in place of sources. The composition instructions generated in Phase 1 ask the synthesis model to credit each child by its node identifier, so that dependencies can be traced, whereas the synthesis prompt asks it to cite sources by number. The two instructions conflict. In the logged run, 4 nodes were composed, including the root, and three of them, the root among them, followed the composition instructions and wrote references such as “(node_3)” after their claims without any source numbers. The report writer therefore receives a root answer that contains no source citations, together with an instruction to cite by number, and it assigns numbers without any basis for choosing them. This explains why the final report of the logged run performs at chance.

8.4 Correction

Both failures disappear if every source keeps a single identity throughout the run. In the corrected pipeline, each source receives one global number when any node first retrieves it, and every leaf answer is rewritten from local to global numbers before it is stored. Children and parents then share one numbering, and the bibliography of the report uses the same numbering. After each composition, node identifiers in a parent’s answer are replaced by the global citations of the corresponding children, and numbers outside the parent’s sources are removed. Both the synthesis prompt and the composition-instruction prompt now require citation by source number and prohibit citation by node. When the logged run is replayed with this numbering, the leaf claims keep their attribution, with the best match cited for 0.96 of claims, the same rate as in the leaves themselves. Whether the report-writing step preserves this attribution end to end can only be established with new runs, which fall outside the scope of this study, but the test above makes that verification inexpensive.

8.5 Implications

The supported-claims rate in Table 2 counted citation markers. It therefore measures how consistently De(ep)Composition attaches citations, which it does more densely than STORM (Section 7), and not whether its claims are supported by the sources they cite. The evaluated reports should not be read as better grounded than the baselines, because the evaluation does not establish this. The same test can be applied to any system that produces inline citations. Neither baseline could be tested here, however, because the STORM exports lacked URLs and the Deep Research reports were not retained.

9 Discussion

What the results support. Three findings hold across the methods used here. First, De(ep)Composition writes decisive reports. The rater found a firm position far more often than in either baseline, and the lexical measurements agree, with fewer hedges on nine of ten motions and more commitment markers on all ten. Second, De(ep)Composition writes compact reports organized around a claim and cites densely within them. Third, its recursive composition, as implemented in the evaluated version, did not preserve the link between each claim and the source that supports it. The rater’s higher scores for supported claims and evidence use reflect the first two findings, namely consistent citation in a compact report, rather than correct attribution.

The stance result depends partly on the prompt. The report prompt instructs the model to argue for the root answer and not to hedge, so a high rate of firm positions would be expected from the prompt alone. The decomposition contributes a root answer in a definite format for the prompt to argue from, but the experiments do not separate these two contributions. The analysis in Section 7 adds a further observation. Titles and executive summaries commit to a position, but half of the conclusions qualify or condition it. A likely

reason is that the final writing step draws on a graph whose branches cover both sides of the question, so the balanced material reappears at the end. Three ablations would separate these effects. The first keeps the graph and removes the stance instruction, the second keeps the instruction and removes the graph, and the third requires the conclusion to agree with the title.

Why provenance fails in recursive composition. A citation number is a pointer into a list of sources. The evaluated pipeline rebuilt that list at every level of the graph but copied the pointers without changing them, so each pointer came to refer to a different source. The same risk arises in any system that composes cited text in stages, for example by summarizing summaries or by merging the outputs of several agents. The error is invisible both to a reader and to a rubric that counts citation markers, because the report still looks well cited. Two design rules prevent it. Each source should carry one identifier for the whole run, and no model should refer to an intermediate result by an identifier that the final text cannot resolve to a source. A lexical audit such as the one in Section 8 offers an inexpensive check that both rules hold, and attribution measures based on entailment models (Rashkin et al., 2023, Gao et al., 2023) offer a stronger one.

Scale and raters. Ten motions and six transcripts are too few for precise estimates. The human scores come from one rater who is also an author and who, although blind to the system, was not blind to the study. The debates were judged by a small model applying a rubric from an unpublished course project, and two rubric measures had to be dropped because their records were unusable. The paired measurements in Section 7 do not depend on a rater, but they are lexical proxies, and the stance labels in Table 5 come from a single reading.

Uncontrolled factors. The three systems differ in their models, search backends, and prompts, so the comparison measures complete systems rather than the effect of decomposition alone. STORM was built to write balanced encyclopedia articles and is evaluated here on a task it was not designed for, so its low count of firm positions reflects its design rather than a failure. Deep Research’s reports were produced by a commercial product whose internals and model version cannot be controlled. A controlled study would hold the language model and the search API fixed across a flat search agent, STORM, and De(ep)Composition, would include the ablations described above, and would measure attribution with the test in Section 8 alongside human judgment.

Cost. A De(ep)Composition run makes on the order of four model calls per leaf and a few per parent (Section 4). Its cost therefore grows linearly with the size of the graph and is several times that of a report written in a single pass. Much of what it retrieves is never used, since only a median of 25% of the listed sources are cited in the body of the report. Removing sources that no claim uses before the final step would shorten the bibliography without changing the argument.

10 Conclusion

De(ep)Composition turns a question into a graph of sub-questions, answers it from the leaves upward under generated composition instructions, extends the graph where answers are incomplete, and writes a report that argues for the result. On ten contested motions and in six automated debates, it produced more decisive and more densely cited arguments than OpenAI Deep Research and STORM, at some cost in fluency and with conclusions that were often more qualified than its titles. An automatic attribution test showed, however, that the citations in its evaluated reports did not point to the sources their claims came from, because citation numbers lost their meaning as answers were composed. Assigning each source a single identifier for the whole run removes this failure. A natural next step is to repeat the comparison with a fixed language model and search API, the stance ablations, more motions and raters, and attribution measured for every system.

References

- Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17682–17690, 2024.
- Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall/CRC, 1993.
- Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6465–6488, 2023.
- Xinyu Hua and Lu Wang. Neural argument generation augmented with externally retrieved evidence. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 219–230, 2018. URL <https://aclanthology.org/P18-1021>.
- Ken Hyland. *Metadiscourse: Exploring Interaction in Writing*. Continuum, London, 2005.
- Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. DSPy: Compiling declarative language model calls into state-of-the-art pipelines. In *International Conference on Learning Representations*, 2024.
- Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. Decomposed prompting: A modular approach for solving complex tasks. In *International Conference on Learning Representations*, 2023.
- John Lawrence and Chris Reed. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818, 2020. URL <https://aclanthology.org/J19-4006>.
- Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative agents for “mind” exploration of large language model society. In *Advances in Neural Information Processing Systems*, volume 36, 2023.
- Nelson F. Liu, Tianyi Zhang, and Percy Liang. Evaluating verifiability in generative search engines. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 7001–7025, 2023.
- OpenAI. Introducing deep research. <https://openai.com/index/introducing-deep-research/>, February 2025.
- Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, 2023.
- Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. Measuring attribution in natural language generation models. *Computational Linguistics*, 49(4):777–840, 2023.
- Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.
- Yijia Shao, Yucheng Jiang, Theodore A. Kanell, Peter Xu, Omar Khattab, and Monica S. Lam. Assisting in writing Wikipedia-like articles from scratch with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6252–6278, 2024. URL <https://aclanthology.org/2024.naacl-long.347>.

-
- Noam Slonim, Yonatan Bilu, Carlos Alzate, Roy Bar-Haim, Ben Bogin, Francesca Bonin, Leshem Choshen, Edo Cohen-Karlik, Lena Dankin, Lilach Edelstein, et al. An autonomous debating system. *Nature*, 591(7850):379–384, 2021. doi: 10.1038/s41586-021-03215-w.
- Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. BrowseComp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025.
- Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6): 80–83, 1945.
- Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023a.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023b.
- Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. Least-to-most prompting enables complex reasoning in large language models. In *International Conference on Learning Representations*, 2023.
- Mingchen Zhuge, Haozhe Liu, Francesco Faccio, Dylan R. Ashley, Róbert Csordás, Anand Gopalakrishnan, Abdullah Hamdi, Hasan Abed Al Kader Hammoud, Vincent Hermann, Kazuki Irie, Louis Kirsch, Bing Li, Guohao Li, Shuming Liu, Jinjie Mai, Piotr Piękos, Aditya Ramesh, Imanol Schlag, Weimin Shi, Aleksandar Stanić, Wenyi Wang, Yuhui Wang, Mengmeng Xu, Deng-Ping Fan, Bernard Ghanem, and Jürgen Schmidhuber. Mindstorms in natural language-based societies of mind. *arXiv preprint arXiv:2305.17066*, 2023.

A Prompts

This appendix reproduces the instructions of the DSPy signatures used for the evaluation runs. Input and output fields are omitted, and lists that span several lines in the original are joined into sentences.

LayerChildGenerationSignature (Phase 1, one layer of children)

Generate children for a specific parent node in the research DAG.
You are a research planning expert. Given a parent research question and the current DAG structure, generate focused sub-questions that can be answered independently to collectively answer the parent question. Look at the entire DAG context to ensure coherence and avoid duplication.

Guidelines:

1. Generate focused, standalone sub-questions that collectively answer the parent
2. Consider the existing DAG structure to maintain coherence
3. Respect the max_children limit strictly
4. Each child should be answerable independently
5. Output format should always be "list" (bullet points)

CompositionInstructionSignature (Phase 1, per parent)

Write explicit parent composition instructions referencing each child node.
Given a parent node and its children, explain exactly how the child answers will be combined to produce the parent's output. Always reference children by their IDs so downstream steps can trace dependencies unambiguously.

LeafNodeResearcher (Phase 2, leaves)

Answer a leaf node research task using literature search results.
Given literature search results and the expected output format, provide an answer that strictly adheres to the format requirements.

Format types and requirements:

- boolean: Provide "Yes." or "No." followed by a single sentence justification with citations.
- short_answer/list/report: Provide 3-5 bullet points ('- ' prefix) summarizing key takeaways with citations.
- table_csv/json: Provide structured data (CSV or JSON) with inline citations in headers or cells.

CRITICAL: Preserve all citations from the literature search in [X] format. Do not add citations that aren't in the source material.

ParentNodeSynthesizer (Phase 2, parents)

Synthesize answers from child nodes to answer a parent node's research task.
Given the answers from all child subtasks and composition instructions, combine them to produce the answer for the parent task in the required format.

CRITICAL RULES:

1. Use ONLY the information from the child results - no external knowledge
2. Follow the composition instructions exactly
3. When citing evidence, use the citation numbers [1], [2], etc. from the AVAILABLE CITATIONS section - do NOT reference child nodes by name
4. Output in the specified format
5. If child results contradict, acknowledge and explain the contradiction

GapExplorer (Phase 2, refinement, first pass)

Explore potential gaps that could enhance the composed answer.
This is the FIRST PASS - be creative and identify any areas where the answer could be improved, even if they're minor enhancements. Don't filter yet.
Think about: missing perspectives or viewpoints. Unaddressed sub-questions or aspects. Areas that could use more depth or detail. Counterarguments or

alternative viewpoints not covered. Quantitative data or evidence gaps. Temporal or geographic coverage gaps. Edge cases or exceptions not mentioned. Return a list of potential gaps, even if they seem minor. The second pass will determine which ones are critical enough to warrant refinement.

GapImportanceEvaluator (Phase 2, refinement, second pass)

Evaluate whether identified gaps are important enough to warrant refinement. This is the SECOND PASS - evaluate gaps based on the specified conservatism level. CONSERVATISM MODES:

- "conservative": Only flag gaps that are CRITICAL - the answer would be misleading or fundamentally incomplete without this information. Default to NOT refining.
- "balanced": Flag gaps that would SIGNIFICANTLY improve the answer's completeness, depth, or accuracy. Be moderate - refine if the gap meaningfully enhances quality.
- "aggressive": Flag gaps that would improve the answer, even if moderately. Refine if the gap would add value, even if not critical.

Consider: Would adding this information fundamentally change the answer's conclusion? Is this gap misleading or would it cause the answer to be wrong? Would a reasonable reader say the answer is incomplete without this? Is this a core aspect of the question that's missing? How much would this gap improve the answer's quality?

Provide a confidence score (0.0-1.0) indicating how confident you are that refinement is warranted.

OutlineFromDAGSignature (Phase 3)

Generate a report outline that structures a persuasive argument from DAG research. The DAG represents a research question decomposed into sub-questions. Each node answered a question through research. The root node has the final answer. Child nodes contain supporting evidence that led to that answer.

Create an outline for a persuasive report arguing a definitive stance for the root answer. Use the DAG's hierarchy to organize evidence: child findings support parent conclusions. Structure as a logical argument (Executive Summary, Evidence Sections, Conclusion).

REQUIREMENTS: Start with Executive Summary stating root answer prominently. Organize body sections to build argument logically using DAG hierarchy. Use ## for main sections, ### for subsections. Design for prose writing (paragraphs, not just lists). Base outline ONLY on content actually present in root_answer (no placeholders).

ReportFromDAGSignature (Phase 3)

Generate a comprehensive final report from the DAG results.

The report must: 1. Start with the root answer prominently (matching its format) 2. Follow the provided outline structure 3. Incorporate findings from all DAG nodes 4. Take a clear stance that matches the root answer 5. Preserve all citations [X] from the DAG 6. Be well-written, coherent, and comprehensive

CRITICAL STANCE ALIGNMENT: If root answer is "Yes" (boolean), the report must argue FOR that position. If root answer is "No" (boolean), the report must argue AGAINST. If root answer is a list, the report should be organized around those items. The report's conclusion must align with the root answer. Do not contradict or hedge against the root answer's stance.

FORMATTING REQUIREMENTS: The report title (# heading) should be the CLAIM, not the question. Use proper markdown. Write in prose paragraphs with supporting bullet points where appropriate. Do not include internal node references in the output. Only use citation numbers like [1], [2], [3]. Every section must contain substantive content from the dag_results. Do not include a bibliography section (it is added later).

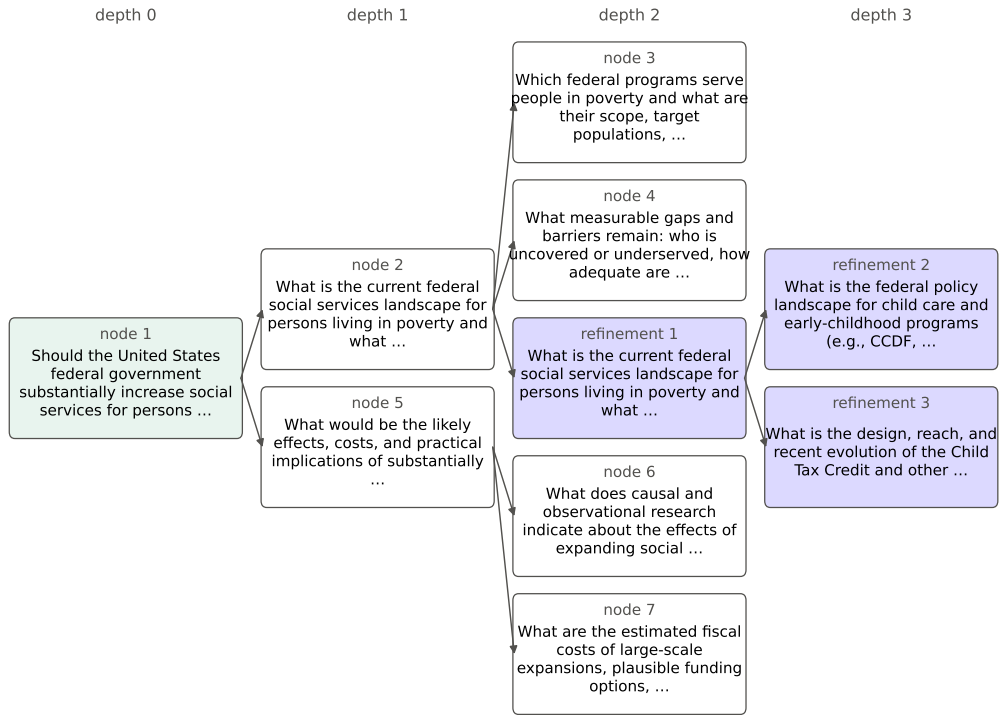


Figure 4: A generated and refined DAG. Arrows point from parent to child. Shaded nodes were added by refinement after the composed answer of node 2 was judged incomplete.

B An Example DAG

Figure 4 shows the DAG from a logged run on the motion “Should the United States federal government substantially increase social services for persons living in poverty?” with the depth capped at 2. The planner divided the root question into the current landscape of federal services and the likely effects of expanding them, and divided each of these into two leaves. After composing the landscape node, the refinement check judged that child care and family tax credits were missing, and the pipeline attached a refinement sub-tree with two new leaves, shown shaded. The 6 leaves of the run retrieved 26 distinct sources, or 4.3 per leaf. Of the 26 sources available at the root, 9 came from the refinement sub-tree, so refinement contributed about a third of the evidence. The final report contains 3154 words and 213 citations and cites all 26 sources, although Section 8 shows that these citations do not match their claims. The full log, with the answer and citations of every node, is included in the repository.

C Code Versions

Three versions of the pipeline are relevant to this study, and two of them are available.

1. **Evaluated version.** The reports in Sections 5 and 7 were finalized on 2 December 2025. The prompts of this version, reproduced in Appendix A, are taken from the authors’ course report. In this version the DAG is generated one layer at a time, composition instructions are written in a second pass, gaps are identified by a two-pass explorer and evaluator, and the bibliography separates cited from uncited sources. The code of this version is not in the public repository.

Motion	De(ep)Composition			STORM	
	words	cited	listed	words	cited
Human creativity will not remain fundamentally irreducible to artificial intelligence when...	1,235	8	31	1,991	32
Individuals have a moral duty to prioritize the well-being of strangers equally to the...	1,523	8	38	1,887	22
Investing in companies developing foundation models -e.g., OpenAI, Anthropic- will yield...	1,247	9	37	2,712	30
Israel and Palestine should pursue a long-term joint economic zone to promote regional...	1,363	9	35	3,763	35
It is ethically justified for hedge funds to use alternative data sources (e.g., satellite...	1,594	10	30	2,797	24
Moral systems rooted in theism are preferable to non-theistic moral systems.	1,393	11	36	2,010	23
The benefits of the African Continental Free Trade Area outweigh the harms	1,195	7	38	2,376	32
The U.S. Federal Reserve should prioritize maintaining high interest rates over stimulating...	1,226	7	32	3,265	24
The U.S. should develop wind, solar, and geothermal energy projects in the Arctic to...	1,315	13	33	2,477	32
U.S. semiconductor manufacturers will outperform global peers over the next decade due to...	1,605	8	39	2,014	23

Table 7: Body length in words and number of distinct sources cited in the body of each retained report, and, for De(ep)Composition, the number of sources listed in its bibliography.

2. **Repository version.** The last commit in the original repository, dated 26 November 2025, generates the whole DAG and its composition instructions in a single call (`FullDAGGenerationSignature`) and checks for gaps with a single validator (`CompositionValidator`) that reports none unless a core aspect of the question is missing. The logged run in Appendix B was produced by this version, and its handling of citations is the one analyzed in Section 8.3. The course report states that citations were later given global numbers, but that change is not present in this version.
3. **Corrected version.** The public repository contains the repository version with a configurable model provider, with unused code removed, and with the citation correction described in Section 8. The correction consists of global citation numbers, the replacement of node references with source citations, revised prompts, and the attribution audit (`utils/citations.py` and `utils/attribution_audit.py`), together with tests that replay the logged run.

D Report Statistics by Motion

Table 7 gives the length and the number of cited sources of each retained report. The medians in Table 3 are computed from these values.

E Stance Labels

The labels in Table 5 were assigned by reading the title and final paragraph of each De(ep)Composition report. The repository records each label together with a short quo-

tation from the final paragraph (data/reports/stance_labels.json), so that every label can be checked against the text.

F Word Lists

The rates in Table 3 count whole-word, case-insensitive matches per 1,000 words of body text. The lists follow the categories of Hyland (2005), adapted to argumentative reports.

Hedges. may, might, could, possibly, perhaps, arguably, potentially, appears, appear, seems, seem, suggests, suggest, likely, unlikely, uncertain, unclear, debated, contested, controversial, depends, depending, can be, to some extent, in some cases, remains open.

Contrast markers. however, on the other hand, conversely, while, whereas, although, though, nevertheless, nonetheless, critics, opponents, proponents, detractors, both sides.

Commitment markers. must, should, clearly, demonstrates, demonstrate, shows, show, proves, establishes, therefore, thus, decisively, undeniably, strongly, essential, necessary, outweigh, outweighs, superior, preferable.

Causal connectives. because, therefore, thus, hence, as a result, consequently, leads to, lead to, led to, results in, result in, resulting in, drives, drive, causes, cause, due to, which means, so that, enables, enable, reduces, increases, raises, lowers.