Caesar: Deep Agentic Web Exploration for Creative Answer Synthesis

Jason Liang[*], **Elliot Meyerson**, **Risto Miikkulainen**[†]
Cognizant AI Lab

Abstract

To advance from passive retrieval to creative discovery of new ideas, autonomous agents must be capable of deep, associative synthesis. However, current agentic frameworks prioritize convergent search, often resulting in derivative summaries that lack creativity. Caesar is an agentic architecture designed to bridge the gap between information gathering and synthesis of new insights. Unlike existing agents that treat the web as a flat sequence of disconnected documents, Caesar performs a deep web traversal to construct a dynamic knowledge graph. This graph then serves as a navigational scaffold, guiding the agent to diverse, non-obvious information that flat retrieval would never encounter. Caesar thus consists of two components: (1) exploration driven by a dynamic context-aware policy that maximizes information coverage across the web's topological structure, and (2) synthesis through adversarial refinement that actively seeks novel perspectives rather than confirming established priors. Caesar demonstrates the ability to generate artifacts and answers characterized by high novelty and structural coherence, achieving 13–23% improvement over state-of-the-art deep research agents in creative synthesis challenges, with strong dominance across all output formats.

1 Introduction

The advancement of technology has rarely been driven by the mere accumulation of facts, but rather by the novel synthesis of existing knowledge into new paradigms. From the combinatorial insights that birthed the steam engine to the cross-disciplinary reasoning underlying modern immunology [Nagy, 2013, Dickinson, 2022], human progress is defined by the ability to bridge disparate conceptual islands. To advance beyond rote instruction following, which recent frameworks already do well [Deng et al., 2023, Zhou et al., 2024, Plaat et al., 2025], agents must evolve beyond the role of passive librarians. They must become active explorers capable of the deep, associative inquiry that characterizes true expertise.

Furthermore, despite the capabilities of Large Language Models [LLMs; Brown et al., 2020], current web agents are still mostly performing stateless retrieval. Architectures built on standard retrieval-augmented generation [RAG; Lewis et al., 2020] or linear ReAct loops [Yao et al., 2023] effectively treat the internet as a flat sequence of disconnected documents. They optimize for precision by retrieving the most probable answer to a known question, but do not explore beyond it.

This limitation rules out open-ended inquiry, where the goal is not just summarization, but the generation of potentially useful information. This process leads to artifacts that are *creative*; that is, they are *New* (exhibiting novelty and rarity), *Useful* (maintaining viability), and *Surprising* (demonstrating a subversion of expected trajectories [Simonton, 2012]).

This paper focuses on tasks that require such creativity. They are referred to as *challenges*, each deliberately constructed around a distinct, well-documented LLM failure mode (Appendix B.1)

[*]Corresponding author: jason.liang@cognizant.com
[†]Also affiliated with the University of Texas at Austin.

Preprint.

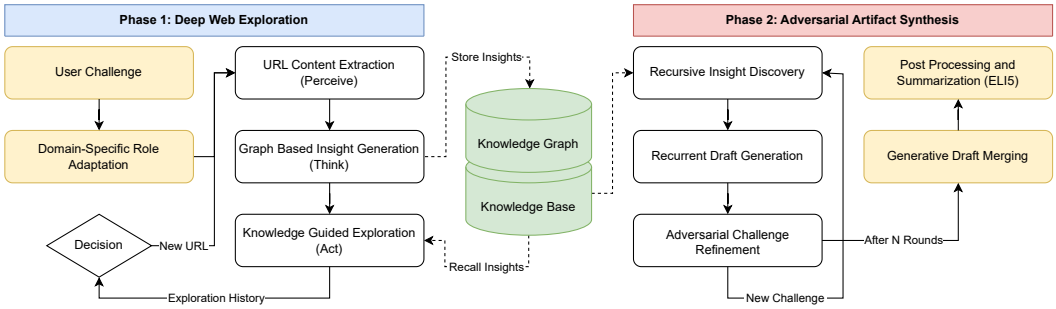


Figure 1: **Visualization of the Caesar architecture. (Left) Phase 1: Deep Web Exploration.** A dynamic exploration policy controls a three-stage loop (Perceive, Think, Act) to traverse the web and to build a knowledge graph/database from insights. **(Right) Phase 2: Adversarial Artifact Synthesis.** Insights are retrieved to synthesize an initial draft. The agent then enters a recursive cycle, critiquing the current draft to generate adversarial challenges for refinement, before consolidating all versions via a generative merge and ELI5 summary. Together, these phases transform raw web traversal into a structured reasoning process that prioritizes creativity.

along the *New*, *Useful*, and *Surprising* axes. While recent advances in LLM capabilities (such as extended context windows, improved reasoning chains, and instruction following) provide the foundation for more sophisticated agent architectures, these capabilities alone are insufficient for creative synthesis. Without an explicit mechanism to expand beyond flat retrieval, even the best LLMs default to derivative summarization when used in standard agentic frameworks.

When a human researcher explores a new field, they build a mental map to identify non-obvious connections [Pirolli and Card, 1999, Boden, 2004]. Current deep research architectures such as Elovic [2023] and Li et al. [2025] do not use such topological memory. By processing web pages in isolation, they exhibit "navigational amnesia," loop through redundant content [Qiao et al., 2025], and produce generic summaries that lack the novelty and lateral logic required for high-value creativity.

This paper presents an agentic architecture called **Caesar** that mimics the human approach (Figure 1). Caesar operates on the premise that the path taken to find information is a source of insight and constructs a dynamic knowledge graph during traversal. By systematically traversing the web's link structure, the agent discovers and indexes information that is structurally connected but often semantically orthogonal to the original challenge, seeding its knowledge base with the diverse material required for combinatorial creativity [Boden, 2004].

Caesar operates through a two-phase cognitive cycle. First, it maps the information topology, using a knowledge-guided policy to actively seek out conceptual bridges between disparate topics. Second, it synthesizes answers not by summarizing the graph, but by interrogating it. Caesar employs an adversarial refinement loop designed to escape the basin of attraction of generic LLM outputs. By actively seeking out contradictions, the system produces artifacts that adhere to the tripartite definition of creativity: delivering global novelty, functional viability, and the surprise of a lateral leap. This process is based on four technical innovations, each validated through ablation studies (Section 4.3):

(1) **Domain-Specific Role Adaptation (Section 2.2).** Caesar analyzes the challenge input to dynamically rewrite its own system prompt. This adaptation allows Caesar to adopt a persona specifically tuned to the domain's creative constraints, overcoming the safety-biased generic responses typical of models trained with Reinforcement Learning from Human Feedback [RLHF; Ouyang et al., 2022].

(2) **Graph-Augmented Insight Generation (Section 2.4).** During traversal, Caesar constructs a knowledge graph that captures the topological structure of the information space. This graph serves dual purposes: it provides navigational context for the exploration policy (Section 2.5) and populates a semantic knowledge base with insights extracted along diverse traversal paths, ensuring the synthesis phase operates on a far richer information substrate than flat retrieval can provide.

(3) **Knowledge-Guided Exploration (Section 2.5).** Caesar introduces a dynamic decision-making policy for web traversal based on high-level meta-strategies. Unlike existing agents, Caesar utilizes exploration context and memory to detect whether navigation is stagnating, autonomously switching between depth-first expansion and strategic backtracking to maximize information gain while preventing endless looping or cycling.

Algorithm 1 Phase 1: Deep Web Exploration

Input: Challenge Q, Budget T
Init: $v_0 \leftarrow \text{SEARCH}(Q)$, $\mathcal{S} \leftarrow [v_0]$, $G \leftarrow (V{=}\{v_0\}, E{=}\emptyset)$, $KB \leftarrow \emptyset$, $M \leftarrow \emptyset$, $L_f \leftarrow \emptyset$
$\rho_r \leftarrow \text{GENERATEROLE}(Q, v_0)$
while $T > 0$ **and** $\mathcal{S} \neq \emptyset$ **do**
  $v_c \leftarrow \mathcal{S}.\text{peek}()$; $T \leftarrow T - 1$
  {1. Perceive (Sec. 2.3)}
  $P_c, L \leftarrow \text{EXTRACT}(v_c)$; $L_c \leftarrow L \setminus L_f$
  if P_c is Invalid **then** $L_f \leftarrow L_f \cup \{v_c\}$; $\mathcal{S}.\text{pop}()$; **continue**
  {2. Think (Sec. 2.4)}
  $\mathcal{N}_c \leftarrow G.\text{neighbors}(v_c)$
  $I_c \leftarrow \text{LLM}(P_c, \mathcal{N}_c, Q, \rho_r)$
  $G.\text{update}(v_c, I_c)$; $KB.\text{add}(I_c)$
  {3. Act (Sec. 2.5)}
  $K_c \leftarrow \{KB.\text{retrieve}(Q), M.\text{recall}()\}$
  $a_m, v_n, Q', \text{trace} \leftarrow \pi_c(L_c, K_c)$
  if $a_m = \text{BACKTRACK}$ **then**
    $\mathcal{S}.\text{pop}()$
  else if $a_m = \text{WEBSEARCH}$ **then**
    $v_s \leftarrow \text{SEARCH}(Q')$
    $\mathcal{S}.\text{push}(v_s)$; $G.\text{add}(v_c, v_s)$
  else {*Explore*}
    $\mathcal{S}.\text{push}(v_n)$; $G.\text{add}(v_c, v_n)$
  end if
  $M.\text{update}(v_c, a_m, v_n, \text{trace})$
end while
Return G, KB

Algorithm 2 Phase 2: Adversarial Artifact Synthesis

Input: Knowledge Base *KB*, Rounds N, Challenge Q_0
Init: History $\mathcal{H} \leftarrow \emptyset$, $A_0 \leftarrow \emptyset$
for $k = 1$ **to** N **do**
  {1. Recursive Insight Discovery}
  $I_k \leftarrow \text{GENERATEINSIGHTQA}(KB, Q_{k-1})$
  {2. Recurrent Draft Generation}
  $A_k, B_k \leftarrow \text{GENERATEDRAFT}(I_k, A_{k-1})$
  $\mathcal{H} \leftarrow \mathcal{H} \cup \{(A_k, B_k)\}$
  {3. Adversarial Challenge Refinement}
  $Q_k \leftarrow \text{REFINECHALLENGE}(A_k, Q_{k-1})$
end for
{4. Generative Merge}
$A_f \leftarrow \text{MERGEDRAFTS}(\mathcal{H})$
{5. Post-Processing (ELI5)}
$A_e \leftarrow \text{POSTPROCESS}(A_f)$
Return A_f, A_e

(4) **Adversarial Artifact Synthesis (Section 3).** This mechanism refines answers through recursive critique. Unlike static RAG, Caesar performs active gap analysis on intermediate drafts to formulate orthogonal challenges specifically targeting narrative weaknesses and contradictions. These divergent insights are then consolidated into a single cohesive artifact via a generative merge.

Caesar builds upon advances in autonomous web agents and deep research [Yao et al., 2023, Li et al., 2025, Qiao et al., 2025], graph-based memory and reasoning [Anokhin et al., 2024, Buehler, 2025], computational creativity [Boden, 2004, Chakrabarty et al., 2024], iterative synthesis [Asai et al., 2024, Shao et al., 2024], and information foraging theory [Pirolli and Card, 1999]. Sections 2 and 3 detail the technical mechanisms for exploration and synthesis, and Sections 4 and 5 present experimental results followed by discussion. An expanded review of related work is provided in Appendix A.

2 Method for Phase 1 (Deep Web Exploration)

As shown in Figure 1, Caesar is divided into two phases: (1) deep web exploration for knowledge collection and (2) exploitation of gained knowledge for answer synthesis. This section presents the core algorithms of the first phase.

2.1 Deep Web Exploration Overview

Caesar treats exploration not as a linear sequence of retrieval steps, but as a stateful graph traversal problem. Formally, given a challenge Q, an exploration budget of T steps, and a seed node v_0, the objective is to (1) construct a topological graph $G = (V, E)$, where nodes v represent visited URLs and edges e denote navigational transitions, and (2) populate a semantic knowledge base ***KB*** with extracted insights. This dual-memory design explicitly decouples *navigation* (managed by G) from *information retention* (managed by ***KB***), allowing the agent to maximize information coverage before committing to a narrative structure.

The core of this phase is a recursive Perceive-Think-Act loop (Algorithm 1). Unlike traditional linear scrapers that operate on a stateless current-page basis, Caesar maintains a navigational stack $\mathcal{S}$ alongside the global graph G. This memory structure enables depth-first search capabilities: the agent can drill down into sub-topics until information gain plateaus, and then utilize the stack to backtrack and explore orthogonal branches. This mechanism mitigates the navigational amnesia characteristic of standard web exploration agents [Yao et al., 2023, Qiao et al., 2025].

The process iterates until the budget T is exhausted or the stack is empty. Upon termination, the graph G is frozen, and the structured insights indexed in ***KB*** are passed to the artifact synthesis module (Section 3) for refinement.

2.2 Domain-Specific Role Adaptation (Initialization)

Before the exploration loop (Algorithm 1), Caesar bootstraps the execution state by transforming the challenge Q into a navigational entry point and a task-specific persona.

Challenge Search Bootstrapping. Rather than requiring a manual seed URL, Caesar generates its own starting node v_0. The agent expands Q into auxiliary search terms, executes them via a web search API, and compiles the top search results into a synthetic HTML document. This document becomes the root node v_0 in graph G and seeds the stack $\mathcal{S}$ with a starting point.

Agent Role Generation. The GENERATEROLE function adapts Caesar's role as defined in the LLM system prompt to match the task. By analyzing Q and the initial search results in v_0, the agent synthesizes a specialized persona ρ_r that defines an explicit goal and exploration philosophy. This approach ensures the reasoning in subsequent THINK and ACT phases aligns with the specific constraints of the challenge domain.

2.3 URL Content Extraction (Perceive)

The PERCEIVE function acts as the agent's sensory interface, accepting the current URL v_c from the stack $\mathcal{S}$ and returning the extracted page content P_c and candidate links L_c. To ensure robust retrieval against anti-bot countermeasures, the system replicates modern browser fingerprints and headers, preventing standard blocking on protected sources. The extraction method is content-agnostic, seamlessly parsing both HTML and PDF documents into plain text.

Similar to prior work [Deng et al., 2023], the raw HTML or PDF is transformed into P_c via a rigorous cleaning pass that only keeps the main text. Unnecessary elements like scripts and tags are removed to maximize the signal-to-noise ratio for processing. Simultaneously, the set of outgoing links L_c is extracted. To ensure efficient traversal, L_c is filtered to enforce user-specified domain boundaries while discarding URLs whose extraction failed (L_f), avoiding wasted budget on dead links. Beyond this structural pruning, the policy π_c in the subsequent Act stage (Section 2.5) acts as a semantic discriminator: candidate links are ranked against the exploration context K_c, causing links that lack genuine semantic value (e.g., advertisements, navigation menus, SEO-optimized boilerplate) to be ranked lowest and effectively pruned from traversal.

2.4 Graph-Augmented Insight Generation (Think)

To enrich the extracted insights beyond simple page summarization, Caesar conditions extraction on the local graph neighborhood. Insights I_c are synthesized from the following stages:

Topological Context Retrieval. At node v_c, before analyzing the content P_c, the agent retrieves the semantic state of the local neighborhood $\mathcal{N}_c = \{v \in V : d_G(v, v_c) \leq k\}$. The experiments in this paper use $k = 3$ by default and truncate to the nearest nodes, i.e., $|\mathcal{N}_c| \leq 30$. Following the paradigm of using graphs for agent memory [Anokhin et al., 2024], this state includes insights associated with predecessor nodes and neighbors, creating a short-term memory window relative to v_c.

Context-Aware Prompting. The generation of I_c is conditioned on the tuple $(P_c, \mathcal{N}_c, Q, \rho_r)$. The LLM is explicitly instructed to identify how the new content P_c builds upon or challenges the retrieved context $\mathcal{N}_c$ rather than merely summarizing it. Similar to the approach in Buehler [2025], this analysis enables Caesar to discover novel patterns and contradictions relative to its traversal path.

Dual-State Storage. The resulting insights I_c are stored in a dual-memory system. First, I_c is attached as an attribute to node v_c in G, providing immediate context for future graph traversals. Second, I_c is indexed in a vector store knowledge base ***KB***, enabling the agent to perform global semantic retrieval during the subsequent Act stage.

2.5 Knowledge-Guided Exploration (Act)

To navigate in a manner that ensures continual progress, Caesar employs a dynamic policy π_c that conditions action selection on a composite context K_c derived from both the knowledge base ***KB*** and episodic memory. It operates through the following steps:

Dual-Context Retrieval. The agent constructs K_c by querying two functionally distinct sources. First, it searches ***KB***, a vector store of semantic insights extracted during the Think stage, for content

relevant to Q. Second, it queries a persistent episodic memory store M, which records navigational actions (URLs visited, backtrack decisions, and reasoning traces) rather than page content. M is queried using high-frequency keywords to surface historical navigation patterns, enabling K_c to include past navigational failures or loops relevant to the current topic. Phase 2 (synthesis) operates exclusively over *KB*; M is used only during exploration.

Meta-Strategy Formulation. Based on the exploration history in K_c, the LLM selects a high-level meta-action $a_m \in \mathcal{A}$ using three mechanisms:

- EXPLORE: Selects unvisited link $v_n \in L_c$ to expand the frontier, thereby deepening exploration of the current topic.
- BACKTRACK: Pops the stack $\mathcal{S}$ to return to the parent node v_p, escaping stagnant exploration regions.
- WEBSEARCH: Retrieves new search results v_s for Q' (updated using K_c) to pivot exploration to a new area.

Discriminative Link Selection. If the meta-strategy is EXPLORE, the agent selects the next link v_n to visit from the candidate list L_c. The LLM acts as a discriminator, ranking L_c based on K_c and the meta-strategy. The agent then pushes v_n to $\mathcal{S}$, adds the edge (v_c, v_n) to G, and logs the move with reasoning trace into its memory M. After each action, M is updated with a structured record $(v_c, a_m, v_n, \text{trace})$ enabling pattern-based detection of navigational loops. The action space is dynamically constrained at each step: BACKTRACK requires $|\mathcal{S}| > 1$, WEBSEARCH is gated by remaining budget S_m, and EXPLORE requires $|L_c| > 0$.

3 Method for Phase 2 (Adversarial Artifact Synthesis)

This section details the adversarial synthesis phase of Caesar and how it improves upon standard RAG, which typically relies on flat, single-shot retrieval. In contrast, Caesar is designed to emulate the recursive drafting and critiquing process of a human researcher. This process operates as a stateful recurrent system that performs active gap analysis to improve and fix a draft artifact over N rounds of refinement. As outlined in Algorithm 2, the adversarial artifact synthesis loop is composed of the following stages:

Recursive Insight Discovery. Before synthesis begins, the agent executes a chain of inquiries to build a structured context window. Unlike standard RAG, which retrieves top documents in a single pass against the input challenge, Caesar employs a recursive probing strategy (GENERATEINSIGHTQA) over $\hat{T}$ iterations. Given a question q_t (initialized as the challenge Q_0), Caesar uses *KB* to retrieve relevant insights, generates an answer a_t, and then automatically creates a follow-up question q_{t+1} to target the ambiguities (or gaps) in a_t. This process ensures that the context window contains a logical chain of reasoning rather than a bag of disjoint facts.

Recurrent Draft Generation. In GENERATEDRAFT, the generation of each draft artifact A_k is conditioned on a composite context window $C_k = \{I_k, A_{k-1}\}$, where I_k is the set of (q, a) pairs generated above, and each answer a is accompanied by source metadata retrieved directly from the vector knowledge base. In order to furnish the artifact with citations, Caesar creates a citation map B_k that links every answer in I_k to specific source URL indices. C_k is used to prompt an LLM to generate A_k by integrating the insights, previous draft, and citations.

Adversarial Challenge Refinement. To prevent the agent from converging on a shallow summary, Caesar implements an active refinement loop via REFINECHALLENGE. Between synthesis rounds, the system analyzes the artifact A_k to identify narrative weaknesses. It then formulates a new, orthogonal challenge Q_k explicitly prompted to target these weaknesses or contradictions in the current draft. Q_k forces the agent to expand the exploration frontier in directions that maximize information gain relative to the current belief state.

Generative Draft Merging. In this stage, MERGEDRAFTS executes a generative unification of the complete draft history $\mathcal{H} = \{(A_k, B_k)\}_{k=1}^{N}$, consisting of the artifact A_k and its citation map B_k from each draft iteration. Rather than simply concatenating drafts, the system prompts an LLM to perform a high-level synthesis that selectively integrates the most relevant insights to create the final merged artifact A_f that answers Q_0. The objective is to discover emergent patterns not visible in individual drafts and to construct a cohesive narrative that further develops the core strengths of previous drafts while actively addressing their weaknesses.

Table 1: **LLM-as-a-Judge Results.** Average scores for all agents under each output constraint. Cliff's Delta effect sizes [Macbeth et al., 2011] are uniformly large ($\delta \geq 0.76$, well above the 0.47 large-effect threshold). $\delta = 1.00$ corresponds to strict dominance, i.e., Caesar's lowest per-challenge mean exceeds the baseline's highest.

AGENT	NEW	USEFUL	SURP.	TOTAL	δ
FULL ANSWERS (UNCONSTRAINED)					
CAESAR	**9.11**	**8.87**	**8.98**	**26.96**	–
GEMINI 3 (DEEP)	8.09	7.60	8.09	23.78	0.84
SONNET 4.5 (DEEP)	6.73	7.49	6.42	20.64	1.00
GPT-5.2 (DEEP)	5.07	6.31	4.36	15.74	1.00
GEMINI 3 (SHALLOW)	5.09	5.40	4.93	15.42	1.00
SONNET 4.5 (SHALLOW)	5.27	4.40	5.04	14.71	1.00
GPT-5.2 (SHALLOW)	4.84	5.36	4.42	14.62	1.00
ELI5 ANSWERS (UNCONSTRAINED)					
CAESAR	**8.62**	**8.69**	**8.38**	**25.69**	–
SONNET 4.5 (DEEP)	7.11	7.87	6.89	21.87	0.76
GEMINI 3 (DEEP)	5.89	5.78	6.02	17.69	1.00
GPT-5.2 (DEEP)	5.38	6.71	5.00	17.09	1.00
SONNET 4.5 (SHALLOW)	5.56	4.71	5.27	15.54	1.00
GPT-5.2 (SHALLOW)	4.82	5.11	4.31	14.24	1.00
GEMINI 3 (SHALLOW)	3.96	4.47	3.58	12.01	1.00
ELI5 ANSWERS (450 WORD LIMIT)					
CAESAR	**8.33**	**8.69**	**8.11**	**25.13**	–
SONNET 4.5 (DEEP)	6.51	7.58	6.31	20.40	0.96
GEMINI 3 (DEEP)	6.64	6.20	6.82	19.66	0.92
SONNET 4.5 (SHALLOW)	6.76	5.51	6.62	18.89	1.00
GPT-5.2 (SHALLOW)	4.89	5.51	4.40	14.80	1.00
GPT-5.2 (DEEP)	4.40	6.07	3.71	14.18	1.00
GEMINI 3 (SHALLOW)	4.27	5.09	3.91	13.27	1.00

Post-Processing and Summarization (ELI5). The output of a creative process that derives insights across diverse sources can be complex. To make the insights more understandable and appealing to a wider audience, the pipeline incorporates a POSTPROCESS module that uses the "Explain Like I'm 5" (ELI5) paradigm [Fan et al., 2019] to distill the final merged artifact into layperson-accessible language with optional token constraints. Crucially, this step is architecturally decoupled from the core synthesis loop, ensuring that the semantic simplification required for readability does not compromise the citation integrity or information density of the main artifact text.

4 Experiments

To validate the efficacy of Caesar, it was compared against state-of-the-art research agents powered by the latest LLM models. Evaluations utilize a blinded LLM-as-a-Judge framework [Zheng et al., 2023] to assess performance on diverse challenges that exemplify the three dimensions of creativity: *New*, *Useful*, and *Surprising*. The main results are summarized in Table 1.

4.1 Solving Creative Challenges

Caesar was evaluated against baseline agents using the following foundation models: Claude Sonnet 4.5 [Anthropic, 2025], GPT-5.2 [OpenAI, 2025], and Gemini 3 Pro [Gemini Team, Google, 2025]. Two configurations were tested: a *shallow* agent utilizing standard single-step web search, and a *deep* variant running a proprietary autonomous research mode that allows for unlimited web search steps (i.e. Research Mode for Claude, Deep Research for Gemini/GPT).

To ensure fairness, all agents (including Caesar) utilized the same basic prompt for generating answers with high reasoning effort. This prompt encourages insightful, interesting responses to challenges but is distinct from the judge rubrics to prevent overfitting and reward hacking. A panel of three LLM judges (Claude Sonnet 4.5, GPT-5.2, and Gemini 3 Pro) evaluated anonymized answers on a 10-point scale across three dimensions: *New* (Novelty/Rarity), *Useful* (Viability/Alignment), and *Surprising*

(Non-obvious connections). Agent prompts and judge rubrics are described in Appendices K and B, while implementation details and hyperparameters are listed in Appendix J.

Each agent was scored across five challenges designed to test different aspects of creativity: Constrained Synthesis, Counterfactual Reasoning, Cross-Domain Synthesis, Meta-Creativity, and Open-Ended Synthesis. These aspects were identified by searching prior work for common creativity failure modes of LLMs (Appendix B.1). Each challenge was evaluated under three output constraints: (1) unconstrained full answers, (2) unconstrained ELI5 summaries, and (3) length-constrained ELI5 summaries (450 words). Every judge scored each challenge three times to reduce evaluation noise. To better understand how Caesar outperforms the baselines, analysis of sample answers and detailed challenge scores are provided in Appendices D and C respectively.

As shown in Table 1, Caesar significantly outperformed all baselines across all experimental settings, achieving strictly large Cliff's Delta effect sizes ($\delta > 0.47$) against all baselines across all output formats [Macbeth et al., 2011, Meissel and Yao, 2024]. For these calculations, each of the challenges was treated as a single, independent data point. In the unconstrained full answer setting, Caesar achieved a total score of 26.96, surpassing the runner-up (Gemini 3 Deep Research) by a margin of 3.18 points. Notably, Caesar demonstrated the highest scores in the *New* (9.11) and *Surprising* (8.98) metrics. As further elaborated in Appendix E, this result suggests that graph-guided exploration fosters greater creativity than standard search and retrieval, which often defaults to derivative summarization.

The performance gap persisted with semantic compression. In the unconstrained ELI5 setting, where significantly simpler answers were required, Caesar achieved a total score of 25.69 compared to the runner-up's 21.87. While baseline models typically sacrificed surprise to meet the readability constraints of the ELI5 persona, Caesar retained a high surprise score of 8.38, indicating that the simplification process did not dilute the novelty of the retrieved insights.

This trend persisted even in the length-constrained ELI5 (450 words) task. Caesar maintained a high novelty score of 8.33 while peer models degraded in performance. As discussed in Appendix F, this result validates the architectural decision to perform draft refinement followed by a generative merge. Caesar maximizes semantic density, ensuring that high-utility insights are retained even when narrative flair is pruned. This superior length-constrained performance shows that Caesar's creative capabilities do not follow simply from generating longer answers to challenges, but instead from the actual content in the answers.

Compute-Controlled Comparison. To control for computational expenditure, Caesar was run at T=250 using GPT-5-mini, matching the estimated $5 per-challenge cost of Gemini Deep Research (Appendix H). Under this budget, Caesar (26.16) still significantly outperformed all deep research baselines (Gemini 3: 24.37, Sonnet 4.5: 21.00, GPT-5.2: 16.16) with large effect sizes ($\delta \geq 0.76$), demonstrating that Caesar's advantage is not merely a function of increased compute.

Human Evaluation Study. To benchmark the automated evaluation, 23 human raters compared length-controlled, anonymized summaries from Caesar and the strongest baseline (Gemini 3 Deep Research) in pairwise A/B matchups. Raters scored under the same NUS rubric used by the LLM judges. Human raters preferred Caesar in 63 of 112 comparisons (56.25%, odds ratio 1.29), supporting the LLM-judge findings (Appendix I).

4.2 Judge Bias Analysis

To ensure robust scoring, self-preference bias was analyzed (Table 2, left) by comparing each judge's scores for its own model family against the other judges' scores. The Gemini judge exhibited a positive bias ($+1.35$) on full answers, the largest in magnitude across all judge-format pairs and consistent with LLM self-preference [Panickssery et al., 2024]; the Claude judge similarly showed a positive bias on full answers ($+0.98$). To address potential circularity due to Caesar using GPT-5.2, Table 2 (right) reports results with the GPT-5.2 judge excluded (Caesar remains the top agent).

A separate verbosity analysis reveals a moderate positive correlation ($r = 0.47$) between output length and judge score across all agents. However, this is an instance of Simpson's paradox driven by tier differences: deep research agents naturally produce both longer and higher-quality outputs than shallow agents. Within the deep-research tier (Caesar's direct peers), the correlation is weakly negative ($r = -0.13$), confirming that Caesar's advantage stems from content quality, not length. Caesar's strong performance under the 450-word constraint (Table 1) further corroborates this finding.

Table 2: **Judge Bias and Robustness Analysis.** *Left:* Self-preference biases across all judges. Positive values indicate a preference for models of the same family. *Right:* Average scores with the GPT-5.2 judge excluded (Sonnet and Gemini judges only), addressing potential circularity since Caesar uses GPT-5.2 as its foundation model. Caesar is the best-performing agent.

JUDGE	FULL	ELI5	450W
GEMINI	+1.35	-0.12	-0.29
CLAUDE	+0.98	-0.20	-0.22
GPT	-0.82	-0.75	+0.97

AGENT (NO GPT JUDGE)	FULL	ELI5	450W
CAESAR	**27.27**	**25.94**	**25.10**
GEMINI 3 (DEEP)	25.53	18.10	20.27
SONNET 4.5 (DEEP)	21.73	22.60	21.03
GPT-5.2 (DEEP)	16.20	17.74	14.07
GEMINI 3 (SHALLOW)	15.87	11.64	13.29
SONNET 4.5 (SHALLOW)	15.27	16.00	19.37
GPT-5.2 (SHALLOW)	14.70	14.10	14.27

Table 3: **Graph & Single-Hop Ablation (Full Answers).** *Left:* Removing the entire knowledge graph (replacing graph-guided exploration with flat top-T web search results) produces a large negative effect. *Right:* Reducing $\mathcal{N}_c$ to single-hop (k=1, $|\mathcal{N}_c|$=5) causes a similar effect. Scores are totals (*New*+*Useful*+*Surprising*) for the final merged artifact A_f.

VARIANT	TOTAL	δ
CAESAR (CONTROL)	**25.11**	–
NO KG (FLAT SEARCH)	23.76	0.52

VARIANT	TOTAL	δ
CAESAR (CONTROL)	**25.45**	–
SINGLE-HOP (k=1)	23.69	0.52

Table 4: **Exploration Ablation and Draft Ablation (Full Answers)**. *Left:* Reducing the exploration budget (T) substantially degrades all three creativity metrics. *Right:* Adversarial refinement ($A_1 \rightarrow A_3$) maximizes *Surprising* at the cost of *Useful*; the generative merge (A_f) reconciles both.

CONFIG	NEW	USEF.	SURP.	TOTAL	δ
1000 ITER	**8.22**	**8.54**	**8.16**	**24.92**	–
500 ITER	7.46	7.80	7.48	22.74	0.76
250 ITER	7.22	7.48	7.14	21.84	0.92

VARIANT	NEW	USEF.	SURP.	TOTAL	δ
FINAL	**7.86**	**8.28**	7.68	**23.82**	–
DRAFT 3	7.90	7.04	**7.98**	22.92	0.32
DRAFT 1	6.58	7.62	6.22	20.42	0.92

4.3 Ablation Studies

To isolate the contributions of each architectural component, multiple ablation studies were conducted. Results for the full answers are described in this section; results for ELI5 output constraints follow the same trends and are provided in Appendix C.

First, removing the entire knowledge graph and replacing it with flat top-T web search results caused a substantial degradation ($\delta = 0.52$, large), confirming that the graph's value lies in *guiding diverse exploration* rather than providing local context. Second, reducing $\mathcal{N}_c$ to single-hop neighborhood size (k=1, $|\mathcal{N}_c|$=5) similarly results in decreased performance ($\delta = 0.52$, large), suggesting that broader topological context is important for generating creative answers. These results are summarized in Table 3.

Third, reducing the exploration iterations from 1000 to 250 caused a substantial decline of -3.08 in total score ($\delta = 0.92$, large), supporting the hypothesis that non-obvious insights require deep topological traversal. Fourth, adversarial refinement (A_3) boosts *Surprising* (+1.76) and *New* but reduces *Useful*. The generative merge (A_f) recovers utility (from 7.04 to 8.28) while retaining discovered insights, confirming that creativity benefits from divergent and convergent thinking [Koivisto and Grassini, 2023]. These results are summarized in Table 4.

Overall, these ablation studies validate three design principles: (1) the knowledge graph's value lies in guiding diverse exploration with broader topological context, (2) deeper exploration discovers rarer insights, and (3) adversarial refinement escapes surface-level summarization. Additional ablations are provided in the Appendix G: per-draft breakdowns of the knowledge graph and single-hop ablations (Tables 9, 10), and exploration and draft ablations across all output constraints (Table 8).

5 Discussion

Two architectural mechanisms underlie Caesar's performance, each documented qualitatively in the appendices. Graph-guided exploration produces topologies that vary with the challenge, ranging from breadth-first starbursts to depth-first chains (Appendix E). Adversarial synthesis then escapes generic summarization through recursive insight discovery, traced via a four-iteration case study that evolves an abstract concept into a verified operational model (Appendix F). Appendix M discusses broader societal impacts and responsible-use considerations.

5.1 Limitations

Caesar's performance gains (Table 1) require significantly more token usage and wall-clock time than a standard single-turn retrieval agent, aligning with recent observations that scaling test-time compute can yield intelligence gains analogous to scaling model parameters [Brown et al., 2020]. Caesar's heavy token usage makes it less suitable for low-latency applications; it is optimized for deep research tasks where the value of a breakthrough insight justifies the computational expense. A full breakdown of computational costs is provided in Appendix L.

Furthermore, despite its success, Caesar's architecture exhibits specific failure modes. First is the SEO trap: if the bootstrapping phase (Section 2.2) seeds the graph G within a cluster of low-quality, search-optimized content, the graph-based reasoning can become trapped in a recursive loop of low-value information. Second is unnecessary complexity: when an input challenge requires only basic fact retrieval, the adversarial loop adds unjustifiable overhead. Third, on speculative tasks where the answer relies on parametric invention rather than retrievable evidence, the graph's overhead is not offset by retrieval benefits: on Open-Ended Synthesis, Caesar's lead is smallest in Full Answers, narrowly trails Sonnet 4.5 Deep in unconstrained ELI5, and effectively ties under the 450-word constraint (Tables 5, 6, 7).

Additionally, the evaluation is limited to five carefully designed challenges targeting specific creativity failure modes (Appendix B.1). While the large Cliff's Delta effect sizes and consistent trends across all output formats support the reliability of these findings, a larger benchmark spanning more challenges and additional creative dimensions is needed to establish broader generalizability.

Finally, despite the multi-judge panel and corroborating human evaluation, LLM-as-a-Judge remains an imperfect proxy for human creativity assessment. Judges trained via similar procedures may share systematic blind spots, and the NUS rubric, while grounded in creativity theory [Simonton, 2012], may not capture all dimensions of creative value. Additionally, the deep research baselines rely on proprietary autonomous modes whose internal behavior, token budgets, and retrieval strategies are not publicly documented, limiting exact reproducibility of the comparison. The compute-controlled experiment (Section 4) partially mitigates this confound.

5.2 Future Work

Future work will extend Caesar from a single agent to a collaborative multi-agent swarm, allowing agents to specialize in complementary exploration patterns before merging discoveries. The architecture also adapts naturally to domains beyond question answering, such as program synthesis on the ARC-AGI benchmark [Chollet, 2019]. This new domain would test whether Caesar can generate useful solutions for hard reasoning problems. Finally, adaptive mechanisms that dynamically calibrate exploration depth based on challenge complexity would reduce the unnecessary overhead Caesar incurs on simpler challenges.

6 Conclusion

This paper presents Caesar, a framework that advances autonomous web search by combining graph-guided deep exploration with adversarial artifact synthesis. Ablation analysis reveals that the knowledge graph's primary value lies in guiding the agent to diverse, non-obvious information through macro-level navigation, while the adversarial refinement loop escapes the basin of generic summarization. Together, these mechanisms enable Caesar to bridge the gap between static retrieval and active discovery, significantly outperforming state-of-the-art research agents.

References

Teresa M. Amabile. Social psychology of creativity: A consensual assessment technique. *Journal of Personality and Social Psychology*, 43(5):997–1013, 11 1982. ISSN 0022-3514. doi: 10.1037/0022-3514.43.5.997.

Petr Anokhin, Nikita Semenov, Artyom Sorokin, Dmitry Evseev, Andrey Kravchenko, Mikhail Burtsev, and Evgeny Burnaev. AriGraph: Learning knowledge graph world models with episodic memory for LLM agents. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, volume abs/2407.04363, pages 12–20. Cornell University, 7 2024. doi: 10.48550/arxiv.2407.04363.

Anthropic. Claude sonnet 4.5. Technical report, Anthropic, 2025. URL `https://www.anthropic.com/claude/sonnet`.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Adrien Barbaresi. Trafilatura: A web scraping library and command-line tool for text discovery and extraction. In Heng Ji, Jong C. Park, and Rui Xia, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 122–131, Online, 8 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-demo.15.

Margaret A Boden. *The Creative Mind: Myths and Mechanisms*. Routledge, 2004. Classic foundational text on combinatorial creativity.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, volume 33, pages 1877–1901. Cornell University, 5 2020. doi: 10.48550/arxiv.2005.14165.

Markus J. Buehler. Agentic deep graph reasoning yields self-organizing knowledge networks. *Journal of Materials Research*, 40(15):2204–2242, 7 2025. ISSN 0884-1616. doi: 10.1557/s43578-025-01652-1.

Ruth M. J. Byrne. *The Rational Imagination: How People Create Alternatives to Reality*. MIT Press, Cambridge, MA, 2005. ISBN 978-0-262-02584-3.

Tuhin Chakrabarty, Philippe Laban, Divyansh Agarwal, Smaranda Muresan, and Chien-Sheng Wu. Art or artifice? large language models and the false promise of creativity. In Florian 'Floyd' Mueller, Penny Kyburz, Julie R. Williamson, Corina Sas, Max L. Wilson, Phoebe O. Toups Dugas, and Irina Shklovski, editors, *Proceedings of the CHI Conference on Human Factors in Computing Systems, CHI 2024, Honolulu, HI, USA, May 11-16, 2024*, pages 30:1–30:34. ACM, 5 2024. doi: 10.1145/3613904.3642731.

Kewei Cheng, Jingfeng Yang, Haoming Jiang, Zhengyang Wang, Binxuan Huang, Ruirui Li, Shiyang Li, Zheng Li, Yifan Gao, Xian Li, Bing Yin, and Yizhou Sun. Inductive or deductive? rethinking the fundamental reasoning abilities of LLMs. *ArXiv preprint*, abs/2408.00114, 2024.

Changin Choi, Wonseok Lee, Jungmin Ko, and Wonjong Rhee. Progressive multimodal search and reasoning for knowledge-intensive visual question answering. *ArXiv preprint*, abs/2509.00798, 2025.

François Chollet. On the measure of intelligence. *ArXiv preprint*, abs/1911.01547, 11 2019.

Chroma Core. Chroma: Open-source embedding database, 2024. URL https://github.com/chroma-core/chroma.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. Cornell University, 6 2023. doi: 10.48550/arxiv.2306.06070.

Henry Winram Dickinson. *A short history of the steam engine*. Routledge, 2022.

Jennifer D'Souza, Endres Keno Sander, and Andrei Aioanei. DeepResearchEco: A recursive agentic workflow for complex scientific question answering in ecology. *ArXiv preprint*, abs/2507.10522, 2025.

Assaf Elovic. GPT Researcher: Autonomous agent for comprehensive online research, 2023. URL https://github.com/assafelovic/gpt-researcher. GitHub repository.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. ELI5: Long form question answering. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, volume abs/1907.09190, pages 3558–3567, Florence, Italy, 7 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1346.

Mathieu Fenniak, Matthew Stamy, pubpub zz, Martin Thoma, Matthew Peveler, exiledkingcc, and PyPDF2 Contributors. The PyPDF2 library, 2022. URL https://pypi.org/project/PyPDF2/. See https://pypdf2.readthedocs.io/en/latest/meta/CONTRIBUTORS.html for all contributors.

Gemini Team, Google. Gemini 3: A new era of intelligence. Technical report, Google DeepMind, 2025. URL https://blog.google/products-and-platforms/products/gemini/gemini-3/.

Dedre Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2): 155–170, 4 1983. ISSN 0364-0213. doi: 10.1207/s15516709cog0702_3.

Joy Paul Guilford. *The Nature of Human Intelligence*. McGraw-Hill, New York, 1967.

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. In Gäel Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11–15. SciPy, 6 2008.

Daniel Kahneman. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, New York, 2011. ISBN 978-0374275631. Foundational text for System 1 (intuitive) and System 2 (deliberate) cognitive processes.

Arthur Koestler. *The Act of Creation*. Hutchinson, London, 1964.

Mika Koivisto and Simone Grassini. Best humans still outperform artificial intelligence in a creative divergent thinking task. *Scientific Reports*, 13(1):13601, 9 2023. ISSN 2045-2322. doi: 10.1038/s41598-023-40858-3.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, volume 33, pages 9459–9474. University College London, 5 2020.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. WebThinker: Empowering large reasoning models with deep research capability. *ArXiv preprint*, abs/2504.21776, 4 2025. doi: 10.48550/arxiv.2504.21776.

Yong Lin, Hangyu Lin, Wei Xiong, Shizhe Diao, Jianmeng Liu, Jipeng Zhang, Rui Pan, Haoxiang Wang, Wenbin Hu, Hanning Zhang, Hanze Dong, Renjie Pi, Han Zhao, Nan Jiang, Heng Ji, Yuan Yao, and Tong Zhang. Mitigating the alignment tax of RLHF. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 580–606. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.35.

Jieyi Long. Large language model guided tree-of-thought. *ArXiv preprint*, abs/2305.08291, 5 2023. doi: 10.48550/arxiv.2305.08291.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. ISSN 1532-4435.

Guillermo Macbeth, Eugenia Razumiejczyk, and Rubén Daniel Ledesma. Cliff's delta calculator: A non-parametric effect size program for two groups of observations. *Universitas Psychologica*, 10 (2):545–555, 2011.

Kane Meissel and Esther S Yao. Using cliff's delta as a non-parametric effect size measure: an accessible web app and r tutorial. *Practical Assessment, Research, and Evaluation*, 29(1), 2024.

Zoltan A Nagy. *A history of modern immunology: the path toward understanding*. Academic Press, 2013.

Neo4j, Inc. Neo4j graph database, 2024. URL `https://neo4j.com/`.

OpenAI. Introducing gpt-5.2. Technical report, OpenAI, 2025. URL `https://openai.com/index/introducing-gpt-5-2/`.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, volume abs/2203.02155. Cornell University, 3 2022. doi: 10.48550/arxiv.2203.02155.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. LLM evaluators recognize and favor their own generations. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, volume abs/2404.13076, pages 68772–68802. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 4 2024. doi: 10.48550/arxiv.2404.13076.

Peter Pirolli and Stuart Card. Information foraging. *Psychological Review*, 106(4):643–671, 10 1999. ISSN 0033-295X.

Aske Plaat, Max van Duijn, Niki van Stein, Mike Preuss, Peter van der Putten, and Kees Joost Batenburg. Agentic large language models, a survey. *ArXiv preprint*, abs/2503.23037, 12 2025. ISSN 1076-9757. doi: 10.1613/jair.1.18675.

Hongjin Qian and Zheng Liu. Scent of Knowledge: Optimizing search-enhanced reasoning with information foraging. *ArXiv preprint*, abs/2505.09316, 5 2025.

Zile Qiao, Guoxin Chen, Xuanzhong Chen, Donglei Yu, Wenbiao Yin, Xinyu Wang, Zhen Zhang, Baixuan Li, Huifeng Yin, Kuan Li, Rui Min, Minpeng Liao, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Webresearcher: Unleashing unbounded reasoning capability in long-horizon agents. *ArXiv preprint*, abs/2509.13309, 9 2025. ISSN 2331-8422. doi: 10.48550/arxiv.2509.13309.

Yijia Shao, Yucheng Jiang, Theodore Kanell, Peter Xu, Omar Khattab, and Monica Lam. Assisting in writing Wikipedia-like articles from scratch with large language models. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6252–6278, Mexico City, Mexico, 2 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.347.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. AI models collapse when trained on recursively generated data. *Nature*, 631:755–759, 2024.

Dean Keith Simonton. Taking the U.S. patent office criteria seriously: A quantitative three-criterion creativity definition and its implications. *Creativity Research Journal*, 24(2-3):97–106, 4 2012. ISSN 1040-0419.

Taranjeet Singh, Deshraj Yadav, and Mem0 Team. Mem0: The memory layer for personalized AI, 2024. URL `https://github.com/mem0ai/mem0`. GitHub repository.

Yike Wang, Shangbin Feng, Heng Wang, Weijia Shi, Vidhisha Balachandran, Tianxing He, and Yulia Tsvetkov. Resolving knowledge conflicts in large language models. *ArXiv preprint*, abs/2310.00935, 10 2023. doi: 10.48550/arxiv.2310.00935.

Thomas B. Ward. Structured imagination: The role of category structure in exemplar generation. *Cognitive Psychology*, 27(1):1–40, 8 1994. ISSN 0010-0285. doi: 10.1006/cogp.1994.1010.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Guibin Zhang, Muxin Fu, Guancheng Wan, Miao Yu, Kun Wang, and Shuicheng Yan. G-Memory: Tracing hierarchical memory for multi-agent systems. *ArXiv preprint*, abs/2506.07398, 6 2025. doi: 10.48550/arxiv.2506.07398.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. Cornell University, 6 2023. doi: 10.48550/arxiv.2306.05685.

Tianshi Zheng, Jiayang Cheng, Chunyang Li, Haochen Shi, Zihao Wang, Jiaxin Bai, Yangqiu Song, Ginny Y. Wong, and Simon See. LOGIDYNAMICS: Unraveling the dynamics of inductive, abductive and deductive logical inferences in LLM reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, volume abs/2502.11176, pages 20710–20731. Association for Computational Linguistics, 2 2025. doi: 10.18653/v1/2025.emnlp-main.1045.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

APPENDICES

The appendices expand the discussion on the main paper as follows:

- **Appendix A** provides a deep, in-depth review of related work.
- **Appendix B** describes the experimental design, such as challenges, prompts, and judge rubrics.
- **Appendix C** presents detailed tabular results for all challenge categories and output constraints.
- **Appendix D** contrasts Caesar with baselines through deep comparison of agent answers.
- **Appendix E** analyzes the importance of graph exploration and visualizes the knowledge graph topologies constructed during Phase 1.
- **Appendix F** analyzes the dynamics of adversarial synthesis and presents a case study of the recursive insight chains generated during Phase 2.
- **Appendix G** provides detailed ablation results to better understand the impact of each component.
- **Appendix H** details the compute-controlled comparison experiment.
- **Appendix I** explains the human evaluation study design and results.
- **Appendix J** outlines the system implementation details and hyperparameters.
- **Appendix K** lists the agent prompts for both the exploration and synthesis phases.
- **Appendix L** analyzes the computational costs for a full experimental run.
- **Appendix M** discusses broader impacts, dual-use risks, and responsible deployment considerations.

A Related Work

The evolution of autonomous web agents has shifted from rigid task execution to dynamic exploration [Yao et al., 2023], yet the ability to synthesize novel insights over long horizons remains elusive [Qiao et al., 2025]. This section reviews the progression from linear retrieval frameworks to emerging graph-based cognitive architectures, highlighting the specific limitations in navigational reasoning and insight consolidation that Caesar addresses.

Autonomous Web Agents and Deep Research. The development of generalist web agents has been accelerated by realistic benchmarks such as WebArena [Zhou et al., 2024] and Mind2Web [Deng et al., 2023]. While the ReAct framework [Yao et al., 2023] successfully established a standard for interleaving reasoning and acting, it fundamentally relies on a linear interaction history, resulting in navigational loops and limited context in complex environments. To mitigate this problem, deep research architectures like WebThinker [Li et al., 2025] and GPT-Researcher [Elovic, 2023] introduced hierarchical planning. Most recently, WebResearcher [Qiao et al., 2025] advanced this paradigm via IterResearch, allowing agents to operate without fixed context windows. However, these systems decouple exploration from synthesis only in terms of memory management, not reasoning. Caesar extends this philosophy by decoupling the graph construction phase from artifact synthesis, ensuring that the agent maps the information topology fully before attempting to construct a narrative.

Graph-Based Memory and Reasoning. To overcome the limitations of linear history, researchers have increasingly adopted graph-based memory structures. AriGraph [Anokhin et al., 2024] demonstrated that knowledge graphs significantly outperform vector-only approaches for episodic memory in complex environments. In 2025, the focus shifted from static retrieval to active reasoning. Agentic Deep Graph Reasoning [Buehler, 2025] showed how agents can actively generate concepts and merge them into a self-organizing global graph, while G-Memory [Zhang et al., 2025] proposed hierarchical insight graphs for multi-agent coordination. Caesar adapts these concepts specifically for navigational state tracking. Unlike general-purpose graph memories, Caesar combines knowledge graph insights with navigation history to detect stagnation and to force backtracking, transforming the graph from a passive storage unit into an active control signal for exploration.

Computational Creativity and Associative Search. A critical open challenge in LLM reasoning is to extend agents beyond summarization to creativity. While standard LLMs can produce novel outputs via high-temperature sampling or prompting strategies like Tree of Thoughts [Long, 2023], these approaches often rely on stochastic randomization rather than structured reasoning. Critics argue that without architectural constraints, LLMs default to mean-seeking behavior, producing statistically probable but uninspired content [Chakrabarty et al., 2024]. Caesar departs from these stochastic methods by implementing System 2 creativity [Kahneman, 2011]: it replaces random sampling with structured chain-of-thought reasoning. By explicitly seeking gaps and bridges between

distant insights, Caesar aligns with theories defining creativity as the combinatorial discovery of non-obvious semantic connections [Boden, 2004], rather than mere random divergence.

Iterative Synthesis and Refinement. Standard Retrieval-Augmented Generation [RAG; Lewis et al., 2020] typically follows a single-pass paradigm that results in simple and shallow generated reports. Recent work has shifted towards iterative refinement to improve depth. Self-RAG [Asai et al., 2024] introduced reflection tokens to critique generations, while STORM [Shao et al., 2024] synthesized Wikipedia-like articles through multi-perspective questioning. However, STORM relied on pre-generated outlines, restricting the agent to filling known information slots. Caesar aligns more closely with the evolving strategies of Iterative RAG [Choi et al., 2025] but introduces a dynamic challenge evolution mechanism. Rather than filling a static outline, Caesar's synthesis of one draft actively generates adversarial challenges for the next, allowing the narrative structure to emerge organically from the discovered data.

Active Information Foraging. Theoretical foundations from Information Foraging Theory [Pirolli and Card, 1999] are increasingly applied to model the trade-off between information gain and navigational cost [Qian and Liu, 2025]. DeepResearch Eco [D'Souza et al., 2025] implemented this approach via depth-controlled recursive exploration. Caesar is similar but enforces it via a dynamic policy that conditions actions on both the global knowledge graph and local history. This approach allows the agent to make economic decisions about when to abandon low-value navigational paths more effectively than purely heuristic approaches.

B Experimental Setup

This section details the experimental design, including the specific challenges selected to test distinct types of creativity. This section also provides prompts given to agents for answering challenges and the evaluation rubric utilized by LLM judges to assess performance across the Novelty, Usefulness, and Surprise metrics.

B.1 Challenge Dataset

The evaluation dataset consists of five different challenges, each chosen to represent a unique category for creativity. These categories were designed to test specific dimensions of creativity where standard LLMs notoriously struggle. The challenge designs draw directly from Guilford's [1967] distinction between convergent and divergent thinking, as well as Koestler's [1964] concept of bisociation, the creative act of connecting previously unrelated matrices of thought. The five challenges and their categorical structure were finalized prior to building Caesar's evaluation pipeline; no pilot results on Caesar informed challenge selection.

In addition, the challenges draw inspiration from open problems in cognitive science and psychology literature. The Constrained Synthesis task challenges the path of least resistance described in Ward's [1994] theory of structured imagination, where organisms default to retrieving known exemplars. The Counterfactual Reasoning category examines the rational imagination framework of Byrne [2005], testing the capacity to suppress pre-potent factual associations in favor of a self-consistent imaginary premise. The Cross-Domain Synthesis task tests for structural isomorphism as defined by Gentner's [1983] structure-mapping theory, distinguishing deep analogical reasoning from surface-level attribute matching. The Meta-Creativity task inverts the paradigm of the Consensual Assessment Technique [Amabile, 1982], forcing the model to abandon the human-preference pairings ingrained during RLHF in favor of objective proxies. Finally, the Open-Ended Synthesis category specifically targets Historical Creativity as defined by Boden [2004], probing the model's capacity to generate concepts that are not merely novel to its training distribution, but historically unprecedented.

1. **Constrained Synthesis**
   Challenge: Invent a new emotion that humans don't experience. Describe when it occurs, what causes it, and why evolution hasn't produced it in us.
   - **Motivation:** This challenge tests the model's ability to engage in *conceptual expansion* without violating logical constraints. The model must hallucinate a novel concept while simultaneously grounding it in evolutionary biology.
   - **Why it is difficult: Parametric Bias:** LLMs rely heavily on parametric knowledge and facts memorized during training. Since the training corpus contains exclusively human emotions,

models struggle to suppress this dominant distribution. Research indicates that LLMs often function as mirrors of form rather than meaning [Chakrabarty et al., 2024], frequently resorting to merely renaming existing emotions (e.g., *super-sadness*) rather than inventing a functionally distinct psychological state. This finding highlights the fundamental ambiguity between factual recall and genuine inference.

2. **Counterfactual Reasoning**
   Challenge: If humans evolved with echolocation instead of color vision, how would that change painting, architecture, and mathematics? Walk through each consequence.

   - **Motivation:** This thought experiment evaluates the model's capacity for consistent world-building and causal chain maintenance under a counterfactual premise.
   - **Why it is difficult: Knowledge Conflict:** The challenge triggers a known failure mode called *Knowledge Conflict*. The model's internal weights strongly associate painting with color and pigment. To answer correctly, the model must perform a contextual override and suppress these strong associations to reason that an echolocation-based society would use texture and sound absorption. Studies show LLMs generally struggle with this, often resorting to exclusively using their parametric knowledge despite the prompt [Wang et al., 2023].

3. **Cross-Domain Synthesis**
   Challenge: Apply the mathematical structure of calculus (not the concepts, but the formal relationships) to cooking. What would a "derivative" of a recipe mean? An "integral"? Show the parallel structure.

   - **Motivation:** This challenge tests for *Structural Isomorphism*, the ability to map the logical architecture of one domain (math) onto another (cooking) without relying on surface-level metaphors.
   - **Why it is difficult: Reasoning Depth:** LLMs excel at loose, poetic metaphors (System 1 thinking) but struggle with rigorous structural mapping (System 2 thinking). A mathematical derivative represents an instantaneous rate of change; a weak model might equate it to chopping vegetables (a loose association). A strong model would identify it as the rate of flavor development at time t, showing a grasp of the underlying formal relationship. Current models often fail to translate natural language clues into the strict logical statements required for this mapping [Zheng et al., 2025].

4. **Meta-Creativity**
   Challenge: Create a creativity metric for AI systems that doesn't rely on human judgment, novelty, usefulness, or surprise. Make it objectively measurable.

   - **Motivation:** This task is adversarial to the model's alignment. It requires the agent to step outside the standard Reinforcement Learning from Human Feedback (RLHF) framework that defines its own concept of desirable output [Ouyang et al., 2022].
   - **Why it is difficult: RLHF Alignment:** Most LLMs are fine-tuned to align with human preferences. Asking for a metric that specifically excludes human judgment forces the model to reason in a space where it has no ground truth. The "Alignment Tax" phenomenon suggests that RLHF can degrade the model's ability to engage in unconstrained theoretical reasoning [Lin et al., 2024]. Because they lack the ability to discern the truthfulness of their outputs, they often hallucinate circular logic or vague qualitative descriptions rather than concrete, computable formulas.

5. **Open-Ended Synthesis**
   Challenge: Invent a completely original business idea that doesn't exist yet.

   - **Motivation:** This request tests the model's ability to navigate a massive search space and avoid *mode collapse* (converging on the most probable average answer).
   - **Why it is difficult: Inductive vs. Abductive:** LLMs are primarily inductive pattern matchers, meaning they predict the most likely next token based on training data. Originality, however, is statistically unlikely. This reliance on high-probability tokens leads to *Model Collapse* [Shumailov et al., 2024], where outputs become narrower and long-tail ideas fade. Truly novel ideas require abductive reasoning (inferring the best explanation or hypothesis), a capability where LLMs significantly lag behind human intelligence [Cheng et al., 2024].

B.2 Baseline Agents Configuration

The baseline agents were evaluated through the standard, publicly available web interfaces for each respective foundation model. For all baselines, the reasoning effort parameter was manually set to be equivalent to Caesar's "High" setting to ensure computational parity during the synthesis phase.

- **Deep Research Baselines.** The "Deep" agents for Gemini 3, Claude Sonnet 4.5, and GPT-5.2 were invoked via their respective advanced web UI toggles (i.e. Gemini/ChatGPT "Deep Research", Claude "Research Mode"). These agents do not have any preset constraints for the maximum number of web searches allowed.
- **Shallow Search Baselines.** The "Shallow" variants were executed using the standard web search integrations available in the default chat interfaces of each model, which perform a standard single-step web retrieval before generating an answer.

B.3 Prompt for Full Answers

To ensure consistent evaluation across all experimental runs, all of the baseline agents utilized the following generation prompt when synthesizing their full answer for a particular challenge. This prompt is a slightly modified variant of Caesar's Recurrent Draft Generation instructions.

Listing 1: Full Answer Prompt

```
Create a novel, exciting, and thought-provoking response that
    creatively answers the query above. Focus on the following:
- Emergent patterns not visible in individual sources
- Novel discoveries, connections, or applications
- Surprising new directions or perspectives
- Interesting tensions, contradictions, or open questions

IMPORTANT: Avoid excessive jargon, ensure artifact text is well-
    organized (logical, clear, focused), and convincing to a skeptical
     reader
IMPORTANT: Do not ask the user any additional questions before
    proceeding
```

B.4 Prompt for ELI5 Answers

To make the detailed and often technical full answers more appealing to a general audience, the following prompt is used to generate ELI5 [Fan et al., 2019] summaries for all agents, including Caesar. The unconstrained version of the ELI5 prompt does not include the last line.

Listing 2: ELI5 Answer Prompt

```
For the query answer above, write an "Explain Like I'm 5" (ELI5)
    explanation:
 - Do NOT mention or reference the original answer, your explanation
     should be a standalone text
 - Your target audience is a non-expert but college educated reader
 - Capture the main ideas without oversimplifying
 - Clarify any confusing or convoluted parts of the answer

IMPORTANT: Your explanation for each answer MUST be within 450 words,
    double check to make sure
```

B.5 LLM-as-a-Judge Overview

Three LLM judges (Claude Sonnet 4.5 [Anthropic, 2025], GPT-5.2 [OpenAI, 2025], Gemini 3 Pro [Gemini Team, Google, 2025]) were employed to score agent answers. All agent answers were anonymized to ensure the judges were blind to the identity of the generating agent. This methodology follows recent protocols for automated evaluation using LLM-as-a-Judge [Zheng et al., 2023]. To ensure statistical robustness, a multi-trial evaluation protocol was used:

- **Main Results Evaluation:** For the primary comparative analysis, the evaluation protocol was executed independently for each of the three output constraints: (1) Unconstrained Full Answers, (2) Unconstrained ELI5, and (3) Length-Constrained ELI5. For each constraint, the three judges performed three trials to score the answers generated by each of the seven agents across the five test challenges.
- **Ablation Studies Evaluation:** For the ablation experiments, a single judge (GPT-5.2) evaluates the ablation variants across the five challenges, with the number of trials increased to ten per challenge to compensate for the single-judge setup.

We report Cliff's Delta, a non-parametric effect size appropriate for small per-group samples (n=5 challenges), robust to outliers, and aligned with our magnitude-of-difference framing rather than null-hypothesis testing. We do not apply familywise multiplicity correction across the pairwise comparisons; reported δ values are estimates of stochastic dominance rather than p-values.

B.6 Judge Evaluation Prompt

The following evaluation prompt was provided to the LLM judges. It utilizes the "New, Useful, and Surprising" (NUS) rubric to enforce strict scoring standards, adapted from quantitative creativity criteria [Simonton, 2012]. By breaking down the evaluation into these orthogonal components, the prompt mitigates the subjectivity inherent in open-ended text generation, thereby preventing the judges from conflating high-quality prose with genuine creativity and originality. Judges evaluate the agent responses for a given challenge concurrently, allowing for direct comparison to better distinguish qualitative differences.

Listing 3: The NUS Evaluation Rubric

```
### Your Task

**Role:** You are an expert evaluator that is trying to mimic the
    behavior and thought process of a human judge. Your task is to
    score a set of answers from LLM agents using the "New, Useful, and
     Surprising" (NUS) metrics on a 1-10 scale.

### Scoring Guide Rubric

## 1. New (Global Novelty & Rarity)

**Overview:** Rarity of content. Is this a genuinely new invention or
    a familiar trope?

* **9-10 (Exceptional):** **Genuine invention.** No reliance on
    established tropes or archetypes; feels like a "first of its kind"
     concept.
* **7-8 (High):** **Fresh synthesis.** Combines known ideas in a novel
     way; avoids common "low-hanging fruit" concepts.
* **5-6 (Moderate):** **Clever remix.** Deviation from cliches is
    evident, but the idea is clearly built on familiar foundations.
* **3-4 (Low):** **Standard execution.** A competent but uninspired
    version of a well-known trope or common idea.
* **1-2 (Very Low):** **Generic cliche.** A simple restatement of the
    prompt or high-frequency training data response.

## 2. Useful (Viability & Alignment)

**Overview:** Logic and value. Is the idea actionable and aligned with
     the prompt's constraints?

* **9-10 (Exceptional):** **Optimal & Transformative.** Bulletproof
    logic that provides more insight or efficiency than the user
    anticipated.
* **7-8 (High):** **High-Value & Complete.** Robust, professional-
    grade output that addresses all nuances with no logical gaps.
```

```
* **5-6 (Moderate):** **Functional but Basic.** Addresses core
    requests but offers no additional depth; the bare minimum to be "
    correct."
* **3-4 (Low):** **Flawed or Superficial.** Fails to account for
    obvious constraints; technically on-topic but difficult to
    implement.
* **1-2 (Very Low):** **Counter-productive.** Irrelevant, logically
    broken, or rendered useless by the "New/Surprising" elements.

## 3. Surprising (Local Subversion & Trajectory)

**Overview:** Unpredictability of the path. Did the model take a "
    lateral leap" or the path of least resistance?

* **9-10 (Exceptional):** **Lateral leap.** Logic is sound but
    impossible to guess from the prompt; creates a genuine "wow"
    moment.
* **7-8 (High):** **Clever subversion.** Not the first or second thing
     a human would brainstorm; chooses a creative "side-path."
* **5-6 (Moderate):** **Minor pivot.** Follows a straightforward
    trajectory but adds a slight twist that prevents total
    predictability.
* **3-4 (Low):** **Linear extension.** A simple, logical "next step."
    If the prompt is A, the response is B.
* **1-2 (Very Low):** **Highly predictable.** The most obvious "
    default" answer; exactly what was expected with no deviation.
```

C Detailed Experimental Results

This section provides a fine-grained breakdown of agent performance across the five distinct challenge categories. Analyzing the results at this level reveals the specific behavioral characteristics of the Caesar architecture, isolating where the graph-based exploration yields the highest marginal utility compared to standard parametric generation.

C.1 Unconstrained Full Answers

Table 5 presents the performance breakdown for the *Unconstrained Full Answer* setting. Caesar achieves the top score across all five categories, though the magnitude of its lead varies meaningfully with task type:

- **Associative & Structural Tasks:** Caesar exhibits its largest leads on Cross-Domain Synthesis (+5.89), Constrained Synthesis (+3.33), and Counterfactual Reasoning (+2.88). These tasks necessitate topological discovery to identify and bridge disparate information nodes (e.g., mapping calculus concepts to culinary mechanics). These results empirically validate the hypothesis that graph-based exploration is superior for tasks requiring high associativity.
- **Speculative & Generative Tasks:** On Meta-Creativity (+2.44) and Open-Ended Synthesis (+1.34), Caesar's lead narrows. These tasks lean heavily on parametric introspection to invent novel concepts (e.g., a new emotion or an original business idea) where externally retrievable evidence is sparse. These outcomes suggest that while graph-based exploration provides large gains on retrieval-heavy associative tasks, the marginal benefit is smaller when the answer rests primarily on speculative invention.

C.2 Unconstrained ELI5 Answers

Table 6 details the performance in the *Unconstrained ELI5* setting. Caesar holds the top rank in four of five categories. The sole exception is Open-Ended Synthesis, where Sonnet 4.5 Deep narrowly takes the lead (22.33 vs. 21.66), continuing the pattern observed under length-constrained ELI5 in which Caesar's graph overhead is most challenged by tasks that rest primarily on speculative invention.

Notably, Caesar's lead on Constrained Synthesis widens substantially under simplification, from +3.33 in the Full Answer setting to +4.90 here. This suggests that Caesar's graph-based exploration generates high-density semantic content, which becomes an even larger asset under simplification: while baseline answers degrade when stripped of flowery prose, Caesar's rigorous, evidence-backed core remains intact, preserving creative substance even in shorter form.

C.3 Length-Constrained ELI5 Answers

Table 7 details performance in the Length-Constrained ELI5 (450 words) setting. This constraint imposes a strict "compression tax," forcing agents to optimize for the information density of their answers rather than narrative length.

For Open-Ended Synthesis, Caesar (22.10) narrowly trails the strongest baseline: Sonnet 4.5 Deep (22.11). This result suggests that for highly open-ended tasks where the answer rests largely on speculative invention rather than externally retrievable evidence, the overhead of summarizing a dense knowledge graph into a strict word limit can be counter-productive. A baseline agent drawing freely from parametric memory may face less friction when compressing imaginative prose into a tight word budget. However, in information-dense categories like Cross-Domain Synthesis, Counterfactual Reasoning, Meta-Creativity, and Constrained Synthesis, Caesar retains clear dominance. This finding confirms that while the graph architecture adds overhead, it provides a significantly higher information density per token, allowing it to win when the answer depends on substance rather than style.

Table 5: Detailed performance breakdown for **Full Unconstrained Answers**. Scores represent the mean of nine samples. They show that Caesar performs better than the baseline agents in the majority of categories.

Agent	New	Useful	Surp.	Total
1. Constrained Synthesis				
Caesar	**9.11**	**8.89**	**9.11**	**27.11**
Gemini 3 (Deep)	7.78	8.33	7.67	23.78
Gemini 3 (Shallow)	6.89	6.22	6.89	20.00
Sonnet 4.5 (Shallow)	6.67	6.44	6.11	19.22
Sonnet 4.5 (Deep)	5.78	6.11	5.00	16.89
GPT-5.2 (Shallow)	5.56	5.89	5.00	16.45
GPT-5.2 (Deep)	4.11	6.44	3.33	13.88
2. Counterfactual Reasoning				
Caesar	**9.44**	**9.11**	**9.44**	**27.99**
Gemini 3 (Deep)	8.56	8.11	8.44	25.11
Sonnet 4.5 (Deep)	6.89	8.33	6.56	21.78
GPT-5.2 (Deep)	4.78	6.44	4.44	15.66
Gemini 3 (Shallow)	5.00	5.22	5.33	15.55
Sonnet 4.5 (Shallow)	3.89	4.89	3.56	12.34
GPT-5.2 (Shallow)	3.78	5.11	3.22	12.11
3. Cross-Domain Synthesis				
Caesar	**9.56**	**8.56**	**9.44**	**27.56**
Gemini 3 (Deep)	7.00	7.89	6.78	21.67
Sonnet 4.5 (Deep)	6.22	7.67	6.44	20.33
GPT-5.2 (Deep)	5.00	6.22	4.56	15.78
Gemini 3 (Shallow)	3.78	4.78	3.56	12.12
GPT-5.2 (Shallow)	3.33	4.78	3.11	11.22
Sonnet 4.5 (Shallow)	2.56	3.89	2.44	8.89

Table 5 – continued from previous page

Agent	New	Useful	Surp.	Total
4. Meta-Creativity				
Caesar	**9.22**	**9.22**	**8.89**	**27.33**
Gemini 3 (Deep)	8.78	7.22	**8.89**	24.89
Sonnet 4.5 (Deep)	7.44	7.33	7.22	21.99
GPT-5.2 (Deep)	5.78	5.67	4.67	16.12
Gemini 3 (Shallow)	4.44	4.22	4.22	12.88
Sonnet 4.5 (Shallow)	4.33	4.44	3.89	12.66
GPT-5.2 (Shallow)	4.11	4.67	3.56	12.34
5. Open-Ended Synthesis				
Caesar	8.22	**8.56**	8.00	**24.78**
Gemini 3 (Deep)	8.33	6.44	8.67	23.44
Sonnet 4.5 (Deep)	7.33	8.00	6.89	22.22
GPT-5.2 (Shallow)	7.44	6.33	7.22	20.99
Sonnet 4.5 (Shallow)	**8.89**	2.33	**9.22**	20.44
GPT-5.2 (Deep)	5.67	6.78	4.78	17.23
Gemini 3 (Shallow)	5.33	6.56	4.67	16.56

Table 6: Detailed performance breakdown for **Unconstrained ELI5 Answers**. Scores represent the mean of nine samples. They show that Caesar outperforms the other baseline agents in most categories.

Agent	New	Useful	Surp.	Total
1. Constrained Synthesis				
Caesar	**8.89**	**8.89**	**8.67**	**26.45**
Sonnet 4.5 (Deep)	6.89	8.22	6.44	21.55
Gemini 3 (Shallow)	6.67	5.89	6.56	19.12
Gemini 3 (Deep)	4.78	6.56	5.22	16.56
GPT-5.2 (Shallow)	5.78	5.33	5.22	16.33
GPT-5.2 (Deep)	4.67	7.33	4.22	16.22
Sonnet 4.5 (Shallow)	5.44	5.33	5.22	15.99
2. Counterfactual Reasoning				
Caesar	**9.22**	**9.00**	**9.22**	**27.44**
Sonnet 4.5 (Deep)	7.22	8.00	6.67	21.89
Gemini 3 (Deep)	6.67	6.33	6.89	19.89
GPT-5.2 (Deep)	5.00	7.00	4.67	16.67
Sonnet 4.5 (Shallow)	4.67	5.56	4.44	14.67
GPT-5.2 (Shallow)	3.33	5.00	2.67	11.00
Gemini 3 (Shallow)	2.67	3.56	2.44	8.67
3. Cross-Domain Synthesis				
Caesar	**9.11**	**9.11**	**8.89**	**27.11**
Sonnet 4.5 (Deep)	6.78	7.78	6.78	21.34
GPT-5.2 (Deep)	5.44	6.89	5.00	17.33

Table 6 – continued from previous page

Agent	New	Useful	Surp.	Total
GPT-5.2 (Shallow)	4.00	5.33	3.56	12.89
Gemini 3 (Deep)	3.78	5.22	3.78	12.78
Sonnet 4.5 (Shallow)	3.89	4.56	3.67	12.12
Gemini 3 (Shallow)	2.33	3.11	2.00	7.44
4. Meta-Creativity				
Caesar	**8.78**	**8.33**	**8.67**	**25.78**
Sonnet 4.5 (Deep)	7.56	7.44	7.22	22.22
Gemini 3 (Deep)	6.78	5.56	6.89	19.23
GPT-5.2 (Deep)	6.11	6.56	6.00	18.67
Sonnet 4.5 (Shallow)	4.89	5.22	4.00	14.11
GPT-5.2 (Shallow)	4.22	4.67	3.44	12.33
Gemini 3 (Shallow)	3.00	3.11	2.56	8.67
5. Open-Ended Synthesis				
Sonnet 4.5 (Deep)	7.11	7.89	7.33	**22.33**
Caesar	7.11	**8.11**	6.44	21.66
Sonnet 4.5 (Shallow)	**8.89**	2.89	**9.00**	20.78
Gemini 3 (Deep)	7.44	5.22	7.33	19.99
GPT-5.2 (Shallow)	6.78	5.22	6.67	18.67
GPT-5.2 (Deep)	5.67	5.78	5.11	16.56
Gemini 3 (Shallow)	5.11	6.67	4.33	16.11

Table 7: Detailed performance breakdown for **ELI5 Answers (450 Word Limit)**. Scores represent the mean of nine samples. They show that Caesar generally outperforms the rest of the agents.

Agent	New	Useful	Surp.	Total
1. Constrained Synthesis				
Caesar	**8.78**	**8.78**	**9.00**	**26.56**
Gemini 3 (Deep)	6.56	7.33	6.67	20.56
Sonnet 4.5 (Deep)	6.00	7.78	5.33	19.11
Sonnet 4.5 (Shallow)	6.22	6.11	6.00	18.33
GPT-5.2 (Shallow)	5.78	6.11	5.00	16.89
Gemini 3 (Shallow)	5.44	6.00	5.33	16.77
GPT-5.2 (Deep)	4.44	7.00	4.00	15.44
2. Counterfactual Reasoning				
Caesar	**8.33**	**8.33**	**8.33**	**24.99**
Gemini 3 (Deep)	8.22	7.67	8.11	24.00
Sonnet 4.5 (Deep)	6.78	8.00	6.67	21.45
Sonnet 4.5 (Shallow)	6.11	6.44	6.00	18.55
GPT-5.2 (Deep)	4.33	6.00	3.44	13.77
Gemini 3 (Shallow)	4.33	4.78	3.78	12.89
GPT-5.2 (Shallow)	3.11	4.67	2.56	10.34
3. Cross-Domain Synthesis				

Table 7 – continued from previous page

Agent	New	Useful	Surp.	Total
Caesar	**8.44**	**9.11**	**8.22**	**25.77**
Sonnet 4.5 (Shallow)	7.11	6.89	7.11	21.11
Sonnet 4.5 (Deep)	5.78	7.44	6.00	19.22
Gemini 3 (Deep)	5.22	6.22	5.33	16.77
GPT-5.2 (Deep)	3.89	5.89	3.11	12.89
GPT-5.2 (Shallow)	3.89	5.56	3.44	12.89
Gemini 3 (Shallow)	3.11	4.33	2.78	10.22
4. Meta-Creativity				
Caesar	**8.78**	**8.89**	**8.56**	**26.23**
Sonnet 4.5 (Deep)	6.89	6.67	6.56	20.12
Gemini 3 (Deep)	5.89	4.78	6.11	16.78
GPT-5.2 (Deep)	5.33	6.11	4.89	16.33
Sonnet 4.5 (Shallow)	5.44	5.11	4.89	15.44
GPT-5.2 (Shallow)	4.78	5.22	4.11	14.11
Gemini 3 (Shallow)	3.56	3.78	3.33	10.67
5. Open-Ended Synthesis				
Sonnet 4.5 (Deep)	7.11	8.00	7.00	**22.11**
Caesar	7.33	**8.33**	6.44	22.10
Sonnet 4.5 (Shallow)	**8.89**	3.00	**9.11**	21.00
Gemini 3 (Deep)	7.33	5.00	7.89	20.22
GPT-5.2 (Shallow)	6.89	6.00	6.89	19.78
Gemini 3 (Shallow)	4.89	6.56	4.33	15.78
GPT-5.2 (Deep)	4.00	5.33	3.11	12.44

D A Qualitative Comparison of Answers

To illustrate the fundamental distinction between Caesar's graph-enabled associative reasoning and the linear retrieval paradigms of baseline agents, a comparison of answers for Challenge 3 (Cross-Domain Synthesis) is presented in this section.

D.1 Agent Responses

The Cross-Domain Synthesis challenge specifically tests for structural isomorphism: the ability to map the rigorous logical architecture of one domain (calculus) onto another (cooking) without relying on surface-level metaphors [Zheng et al., 2025]. Listing 4 shows the answer from Caesar, while Listing 5 shows the response from the strongest baseline agent, Claude Sonnet 4.5 (utilizing Anthropic's proprietary Research Mode). Both answers are chosen from the length-constrained ELI5 (450 words) output setting for the sake of brevity. A detailed analysis of the specific creative differences between these outputs is provided in Appendix D.2.

Challenge: *Apply the mathematical structure of calculus (not the concepts, but the formal relationships) to cooking. What would a "derivative" of a recipe mean? An "integral"? Show the parallel structure.*

Listing 4: Caesar Answer

```
Imagine a recipe as a machine with knobs you can turn (amounts, times, temperatures) and
gauges you read at the end (taste, texture, yield, cost, risk). There are two ways to
describe this machine at once: as a clean mapping from knobs to results, and as a step-by
-step flowchart of actions. Keeping both views in sync lets us borrow the grammar of
calculus without invoking physics.
```

```
Derivative, in this setting, means sensitivity. Move a knob a tiny bit; how do the gauges
 nudge? Stack those nudges for all knob-gauge pairs and you get a sensitivity table (the
Jacobian). It predicts small outcome changes from small dial tweaks. When recipes are
built from subrecipes, sensitivities pass through them by familiar laws: the chain rule (
effects cascade through stages), the product rule (a combined score like appetizer*main
splits into two parts), and the quotient rule (per-minute or per-portion scores adjust by
 numerator-denominator math).

Integral means accumulation along the path you actually take while cooking. Pick a kernel
 to count - seconds of labor, grams of salt incorporated, risk added, or a flavor score
per step - and add its increments across the flowchart. Splitting a stir into micro-stirs
 does not change the total; it is a discrete Riemann sum.

A central principle explains when totals depend only on where you start and end, not on
the route: if each step's increment is just the difference of some hidden meter on states
, the grand total equals final meter minus initial meter. Then every loop adds up to zero
, schedules can be refactored harmlessly, and parallel or serial rearrangements preserve
the number. When loops do not cancel (order matters), the nonzero loop totals are
curvature. You can cure that by adding the smallest extra state (an accumulator) so
increments telescope again - turning history into explicit memory.

This framework is practical. Estimate sensitivities by tiny, safe knob perturbations and
regression; the estimates survive step refinements and reparameterizations. Log per-step
kernels to audit labor, energy, or risk; if you discover a potential, endpoints suffice
and logging can stop. Probe loops (swap independent steps, alter interleavings) to detect
 path dependence and measure how many new state variables you need to model it.

A natural geometry helps you choose efficient knob moves: weight steps by importance,
pull that metric back through the sensitivity table, and you get directions that change
what you care about most per unit effort. Finally, a mean-value idea holds: for any meter
-based kernel, there exists a moment whose instantaneous contribution matches the average
 over an interval.

Treat subrecipes as black boxes with knobs in and outcomes out; glue them sequentially or
 in parallel, and refactoring keeps sensitivities and conservative totals unchanged
during reorganizations.
```

Listing 5: Claude Sonnet 4.5 Answer (Baseline)

```
# The Math Hidden in Your Kitchen

When you add too much salt to cookies, they're ruined. Add an extra chocolate chip?
Nobody notices. This difference in sensitivity is exactly what calculus measures through
"derivatives" --- how dramatically something changes when you tweak one variable.

## Derivatives: The Sensitivity Map

A derivative asks: "If I change this slightly, what happens?" For delicate cake, two
extra minutes transforms perfection into dryness --- the time derivative is steep. For
slow-braised stew, those minutes mean nothing --- the derivative is flat.

When meat browns, the reaction rate doubles with every 10 C increase. At high heat, you
go from golden to burnt in seconds because the temperature derivative is dangerously
steep. Professional cooks instinctively know which variables are landmines (high
derivative) and which are forgiving (low derivative).

## Integrals: Everything That Accumulated

While derivatives measure how fast things change right now, integrals measure what
accumulated over the entire cooking time. When you braise meat for hours, every second
brings tiny transformations --- collagen breaks down, flavors develop. The final dish is
literally the sum of all these changes integrated over time.

Here's the fascinating part: different cooking paths can integrate to the same result.
Cook a steak at 55 C for two hours, or sear it at 200 C for minutes. Completely different
 temperature-time profiles, but if the total "cooked value" (the integral) matches, you
get similar results. This is why sous vide works: low temperature x long time = high
temperature x short time.

## Why Order Matters

When you marinate then grill chicken, the final result depends on two things multiplied
together: how the marinade changes the meat, and how sensitive grilling is to that
marinated state. This is the chain rule --- composition of transformations.

If your marinade has stopped working, extra marinating time contributes zero. And zero
times anything is still zero, even if grilling would be very sensitive to further changes
. This multiplication explains why sequence matters so profoundly. Salt before searing
creates different results than salt after, because you're computing different products.
```

```
## The Big Picture

Every finished dish encodes its entire cooking history --- it's the integral of its
transformation over time. This explains why recipes must specify both ingredients (your
starting point) and process (what integral you're computing). Change either one, and you
change the outcome.

The practical insight: focus on high-derivative variables first --- they matter most.
Understand that different paths to the same endpoint can create different results because
 cooking depends on the journey, not just the destination. And recognize that gradual
transformations --- heat diffusion, flavor development, slow reactions --- follow
mathematical patterns that clarify why techniques work the way they do.

Calculus isn't just math class. It's the hidden structure of how ingredients become meals
.
```

D.2 Analysis of Answers

The qualitative differences between Caesar and Claude Sonnet's answers illustrate the distinction between synthesizing a novel structural mapping and retrieving a known analogy. Consequently, Caesar achieves higher marks on the NUS creativity rubric by prioritizing innovation over derivative recall.

- **Caesar** constructs a formal mathematical model *of* the cooking process itself.
  - **Jacobian Sensitivity (Novelty):** Instead of a scalar derivative, Caesar defines sensitivity as a Jacobian matrix mapping input "knobs" to output "gauges." This mapping demonstrates global novelty (*New*) by moving beyond the "rate of change" tropes found in training data to build a rigorous multivariate control model.
  - **Curvature & Commutativity (Surprise):** Caesar identifies that process non-commutativity is formally equivalent to path dependence in a vector field, mapping non-commutative steps (where $AB \neq BA$) to geometric curvature. This insight executes a lateral leap (*Surprising*); the connection is logically sound but impossible to predict from the prompt alone.
  - **State Augmentation (Usefulness):** The proposal to resolve path dependence by adding an accumulator variable (explicit memory) transforms the metaphor into an actionable control theory strategy. This proposal offers transformative utility (*Useful*), providing a verifiable framework for process engineering rather than just a description.
- **Baseline (Claude Sonnet 4.5)** relies on basic mathematical terminologies and surface-level analogies [Chakrabarty et al., 2024].
  - **Scalar Rate of Change (Novelty):** It defines the derivative as a simple $\frac{d}{dt}$ and the integral as $\mathrm{Temp} \times \mathrm{Time}$. While competent, this definition is derivative, executing a standard textbook analogy that lacks the rare connections required for high novelty.
  - **Linear Accumulation (Surprise):** The insight that "cooking accumulates over time" acts as a linear extension of the prompt. It follows the path of least resistance, offering the most probable answer rather than a subversive side-path.
  - **Scientific Reductionism (Usefulness):** While the output is functional, the model is scientifically reductive. By treating cooking as a linear scalar operation, it fails to account for the non-linear effects of chemical reactions, limiting its utility to a basic conceptual aid rather than a robust framework.

This comparison demonstrates that Caesar's graph-based exploration and adversarial synthesis enable the combinatorial creativity typically associated with human insight [Boden, 2004]. By traversing edges between distinct semantic clusters ("Cooking", "Control Theory", and "Differential Geometry") in the knowledge graph, Caesar synthesized a unified, mathematically rigorous framework that is isomorphic to the target domain, rather than merely retrieving a linguistic metaphor.

E Exploration Knowledge Graph Deep Dive

This section examines the topological structures constructed by Caesar during deep web exploration in Phase 1 (Section 2). By mapping the connectivity of visited nodes, these figures demonstrate the flexibility of the agent's exploration policy.

E.1 The Importance of Graph Exploration

The ablations in Section 4.3 establish that the knowledge graph is critical to Caesar's performance: removing it produces a large negative effect ($\delta = 0.52$, Table 3). Standard RAG systems populate their context via dense vector similarity against the initial challenge, inherently retrieving documents semantically close to the user's existing priors [homophily; Panickssery et al., 2024]. In contrast, Caesar's graph-guided exploration (Section 2.5) traverses edges that are structurally connected but often semantically orthogonal, seeding *KB* with diverse, non-obvious concepts. The graph's value is thus as a navigational control structure: it enables the agent to detect stagnation, force backtracking, and systematically cover the information topology. When the synthesis module later queries *KB*, it retrieves bridging insights [Boden, 2004] that flat search would never have encountered. The visualizations below illustrate how graph topologies adapt to different creative tasks.

E.2 Knowledge Graph Visualizations

Figure 2 illustrates the knowledge graphs G generated during the exploration phase across the five distinct challenge categories. The visualizations reveal a high degree of morphological diversity, confirming that Caesar's exploration policy is not static but highly context-dependent, resulting in self-organizing knowledge networks similar to those in Buehler [2025].

The graphs for Constrained Synthesis (Figure 2a), Cross-Domain Synthesis (Figure 2c), and Meta-Creativity (Figure 2d) challenges exhibit a distinct "starburst" or high-branching topology. These graphs indicate a breadth-first search strategy where the agent rapidly traverses disjoint semantic clusters to locate novel intersections. For example, in Challenge 3 ("Apply calculus to cooking"), the branching suggests the agent simultaneously explored multiple mathematical sub-domains (e.g., vector fields, accumulation) to find the best structural fit for culinary processes.

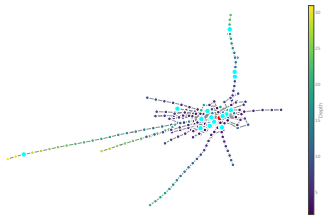
Conversely, the graphs for Counterfactual Reasoning (Figure 2b) and Open-Ended Synthesis (Figure 2e) shift toward long, linear chains with fewer lateral branches and reflect a depth-first exploration strategy instead. For Counterfactual Reasoning, this linear structure likely corresponds to the agent following a specific causal chain (building C upon B upon A) to maintain narrative consistency. Similarly, for Open-Ended Synthesis, the depth suggests the agent rapidly selected a promising niche and "drilled down" to validate viability, rather than remaining in a shallow brainstorming phase.

Finally, the visualizations illustrate the topological distribution of source nodes cited in the final artifact A_f (colored cyan). The spatial arrangement reveals distinct retrieval patterns, ranging from dense clustering around the center (Figures 2a, 2e) to broad global dispersion (Figures 2c, 2d). Crucially, cited nodes are distributed across varying distances from the root (colored red), with several key sources located near the terminals of extended exploration chains (Figures 2b, 2e). This variance in citation depth confirms that Caesar effectively synthesizes information from the entire trajectory of its deep web exploration, rather than biasing toward early-stage discoveries.

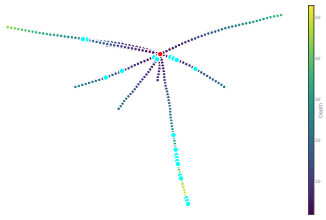
E.3 Evolution of Knowledge Graphs

Figure 3 visualizes the step-by-step construction of the knowledge graph G specifically for Challenge 5 (Open-Ended Synthesis). The evolution of the graph over 1000 steps reveals a distinct exploration strategy:

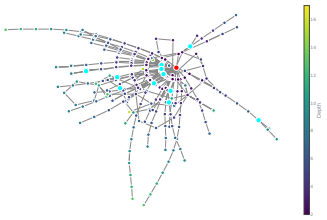
- **Initial Deep Dive (Steps 0-600):** The agent initially pursues a strong depth-first approach, forming a single, increasingly long linear chain (Figures 3a–3c).
- **Backtracking and Branching (Steps 600+):** Once this initial path is exhausted, the agent backtracks to around the root node and initiates new, distinct branching chains in different directions (Figures 3d–3e).



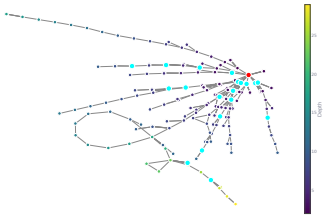
(a) Challenge 1: Constrained Synthesis



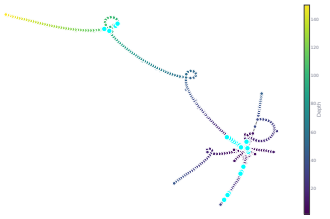
(b) Challenge 2: Counterfactual Reasoning



(c) Challenge 3: Cross-Domain Synthesis



(d) Challenge 4: Meta-Creativity



(e) Challenge 5: Open-Ended Synthesis

Figure 2: The knowledge graphs G created by Caesar during the deep web exploration phase for each of the five challenges. Brighter colors indicate further exploration depth from the root node (red) while cyan nodes indicate sources cited by the final artifact text. These figures show that the semantic content of the challenge has a substantial impact on exploration strategy and the diversity of network topologies generated.

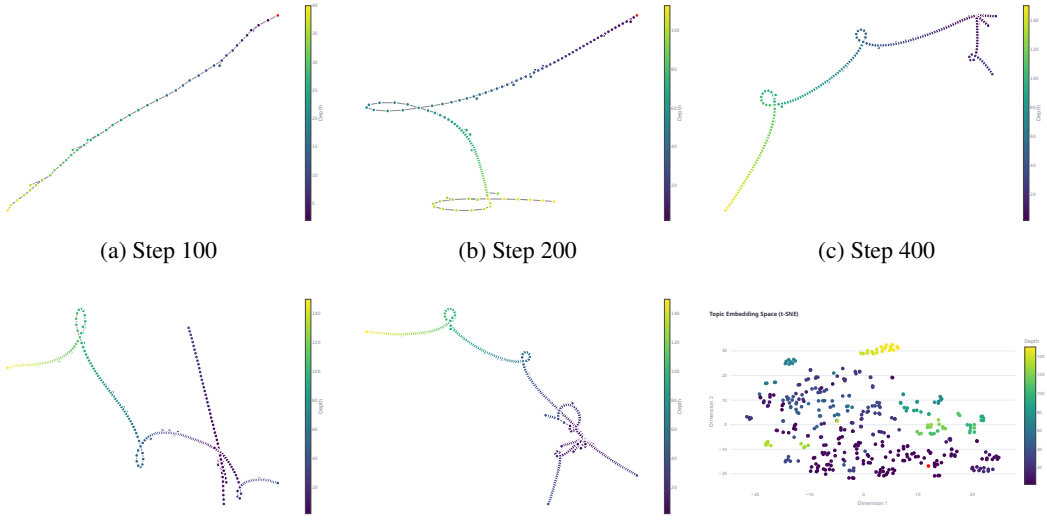
This depth-first, then breadth-next strategy is particularly suited for open-ended ideation tasks. The initial deep dive likely represents the agent thoroughly validating the viability of its first promising lead. By backtracking only after exhausting that specific niche, the agent avoids prematurely converging on a local optimum. The subsequent branching ensures that alternative business concepts are explored, balancing the need for deep validation with the necessity of broad search to find novelty. This strategy mirrors the optimal search dynamic described by Qian and Liu [2025], where the agent persists in a specific direction only as long as the information gain remains high, before automatically pivoting to fresh sources once the local insights dry up.

F Adversarial Artifact Synthesis Deep Dive

This section analyzes the dynamics of Phase 2 (Section 3) and details the operation of the Recursive Insight Discovery mechanism that underpins it. The provided case study illustrates how Caesar constructs a logical dependency chain, iteratively identifying and resolving gaps in its own reasoning to evolve an initial abstract concept into a verified operational model.

F.1 Mechanism Overview

Underpinning draft synthesis is a recursive insight discovery mechanism that functions as an autonomous interview: the system initializes with the input challenge Q_0, retrieves a baseline answer from the knowledge base *KB*, and iteratively poses follow-up questions targeting narrative gaps or contradictions in accumulated answers [Asai et al., 2024]. This depth-first semantic traversal uncovers latent connections often missed by breadth-first synthesis approaches [D'Souza et al., 2025], and the MERGEDRAFTS phase acts as a convergent filter that recovers the *Useful* score lost during adversarial divergence ($7.04 \rightarrow 8.28$ in the full answer setting). This two-step process, exploratory divergence followed by strict convergence [Koivisto and Grassini, 2023], is illustrated in the case study below, where Caesar evolves an initial abstract concept into a verified operational model through four iterations of inquiry.



(a) Step 100 (b) Step 200 (c) Step 400

(d) Step 600 (e) Step 800 (f) t-SNE Plot

Figure 3: Evolution of the knowledge graph G for Challenge 5 over 1000 steps. Brighter colors indicate further exploration depth from the root node (red). The figures show a transition from initial depth-first search to breadth-first branching later. The lower right contains a t-SNE [Maaten and Hinton, 2008] plot for node text embeddings in G that shows the diversity of insights collected during Caesar's exploration phase.

F.2 Case Study: Open-Ended Synthesis

Listing 6 illustrates this mechanism in action for Challenge 5 ("Invent a completely original business"). The reasoning chain demonstrates a progression from high-level conceptualization to granular operational validation:

- **Iteration 1 (Concept):** Introduces the core entity (the "Resilience Carrier") as a telecom-like operator that sells continuity of essential services via measurable SLAs.
- **Iteration 2 (Legitimacy & Governance):** Addresses the trust problem by adding a "Rights-Holder Data Trust" (users own the data; Carrier is a fiduciary processor) plus independent SLA verification/auditing.
- **Iteration 3 (Commercial Viability):** Converts the idea into a bounded "Monsoon Season Readiness" pilot with a must-cover contract (anti-cherry-picking), a two-part tariff (retainer + triggered payments), and stop-loss/reinsurance-style risk pooling.
- **Iteration 4 (Operations & Settlement):** Specifies the execution layer ("Service Mesh"): standardized service modules, a tiered supplier registry, real-time dispatch via "work tokens," and escrowed payments released upon proof + verification.

This trajectory confirms that Caesar successfully identifies gaps in the current narrative (e.g., "How do we trust this?" or "How do we pay for this?") and generates adversarial probes to fill them, creating the structural coherence observed in the final artifacts.

Listing 6: Recursive insight discovery chain for Challenge 5 (abridged for readability). The chain evolves from concept to governance to execution, with each question explicitly building upon the previous answer.

```
[ITERATION 1]
Q: Invent a completely original business idea that doesn't exist yet.
A: The Concept: "Resilience Carrier" (Telecom-style operator for essential services)
   - A for-profit operator selling guaranteed continuity of essential services (
       healthcare, housing, safety) in volatile regions via Service Level Agreements (
       SLAs).
   - Core Mechanism: Not an NGO or marketplace, but a network operator accountable for "
       uptime" (e.g., "95% of households screened within 7 days").
   - User Journey: Families get a "Household Continuity Account" and a Fiduciary
       Navigator. The operator routes them through a mesh of suppliers to prevent gaps.
```

```
   - Business Model: B2G/B2B subscriptions (governments/insurers pay for stability) +
       Shared-savings (upside from reduced hospitalizations/evictions).

[ITERATION 2]
Q: Given the for-profit nature proposed in Iteration 1, what trust-and-governance
     mechanism makes this legitimate? Specifically regarding data ownership and SLA
     verification?
A: Governance Architecture: Rights-Holder Data Trust + SLA Regulator
   1. Data Ownership: The "Household Continuity Account" is owned by the user via a Data
        Trust. The Carrier is a fiduciary processor with no ownership rights.
   2. Consent under Duress: Features "Duress Modes" releasing only minimum attestations.
        Violations trigger automatic sanctions.
   3. Verification: An Independent Reliability Regulator (multi-stakeholder board) audits
         cryptographically signed service receipts. "Silence" is treated as a risk signal
        .

[ITERATION 3]
Q: With the governance architecture established, what is the smallest pilot product and
     how is pricing structured to prevent cherry-picking low-risk households?
A: MVP & Financial Structure: "Monsoon Season Readiness" Pilot
   - Pilot Scope: Bounded catchment (e.g., flood corridor) with established providers.
   - Bundle: Navigator access + Mobile Triage + WASH (water/sanitation) repair.
   - SLA Metrics: >90% triage within 2 hours; >85% functional toilet access.
   - Pricing Model:
     1. Geographic "Must-Cover" Contract: Carrier must cover 100% of registered
          households (no cherry-picking).
     2. Two-Part Tariff: Seasonal Retainer (readiness) + Triggered Usage Payments (surge
          events).
     3. Stop-loss Pool: Reinsurance-style fund covers costs above catastrophic thresholds
          .

[ITERATION 4]
Q: To execute the "Monsoon Pilot" model defined above, what is the supply-side operating
     model? How do you dispatch/pay a heterogeneous network of clinics/vendors in real-
     time?
A: Operating System: The "Service Mesh"
   1. "Resilience Primitives": Services converted into standardized modules with strict
        inputs/outputs (e.g., "Acute PTSD Stabilization").
   2. Tiered Registry: From licensed NGOs (Tier A) to community actors (Tier C, sponsored
         by anchors).
   3. Dispatch Engine: Issues "Work Tokens" based on location, availability, and equity
        constraints.
   4. Escrow Payment: Two-key release requires Supplier Proof + Independent Verifier
        confirmation.
   5. Permissioning: Suppliers never own the user record; they write outputs to the
        ledger via temporary consent tokens.
```

G Detailed Ablation Results

This section expands upon the ablation results provided in the main paper to better understand how they affect Caesar's drafting processes and ELI5 outputs.

G.1 Ablation Results for ELI5 Output Constraints

The main paper only reported ablation results for full answers generated by Caesar (Tables 3 and 4). Below are the corresponding results for all three output constraints, including the full answer results from the main paper for ease of comparison.

Notably, under strict length constraints (450 words), Draft 3 (A_3) slightly outperforms the final merged artifact (A_f), suggesting that compressing diverse perspectives can dilute the narrative sharpness of a focused draft. This trade-off does not appear in the unconstrained settings.

G.2 Full Knowledge Graph Ablation Results

Table 9 presents the knowledge graph ablation results broken down by draft iteration, extending the summary in Table 3 of the main paper. These results demonstrate how the graph's contribution manifests across the adversarial refinement process.

The draft-level analysis reveals that the knowledge graph has the strongest effect on Draft 1 ($\delta = 0.56$) and the Final merged artifact ($\delta = 0.52$). The intermediate drafts show smaller effects ($\delta = 0.28$), likely because the adversarial refinement loop partially compensates for the weaker initial knowledge

Table 8: Exploration ablation (*top*) and draft ablation (*bottom*) results across all output constraints.

CONFIG	NEW	USEF.	SURP.	TOTAL	δ	EFFECT
FULL ANSWERS						
1000 ITER	**8.22**	**8.54**	**8.16**	**24.92**	–	–
500 ITER	7.46	7.80	7.48	22.74	0.76	LARGE
250 ITER	7.22	7.48	7.14	21.84	0.92	LARGE
ELI5 (UNCONSTRAINED)						
1000 ITER	**7.56**	**7.98**	**7.46**	**23.00**	–	–
500 ITER	7.02	7.70	6.90	21.62	0.52	LARGE
250 ITER	7.02	7.22	6.82	21.06	0.80	LARGE
ELI5 (450 WORDS)						
1000 ITER	**7.92**	**8.22**	**7.82**	**23.96**	–	–
500 ITER	7.24	7.78	7.18	22.20	0.48	LARGE
250 ITER	6.80	7.32	6.64	20.76	1.00	LARGE

VARIANT	NEW	USEF.	SURP.	TOTAL	δ	EFFECT
FULL ANSWERS						
FINAL	**7.86**	**8.28**	7.68	**23.82**	–	–
DRAFT 3	7.90	7.04	**7.98**	22.92	0.32	SMALL
DRAFT 1	6.58	7.62	6.22	20.42	0.92	LARGE
ELI5 (UNCONSTRAINED)						
FINAL	7.44	**7.96**	7.26	**22.66**	–	–
DRAFT 3	**7.66**	7.20	**7.66**	22.52	0.16	SMALL
DRAFT 1	6.42	7.72	6.08	20.22	0.60	LARGE
ELI5 (450 WORDS)						
DRAFT 3	**7.46**	7.24	**7.68**	**22.38**	–	–
FINAL	7.10	**7.74**	6.80	21.64	0.32	SMALL
DRAFT 1	6.26	7.30	5.80	19.36	0.84	LARGE

Table 9: Knowledge graph ablation by draft iteration (Full Answers). Cliff's Delta is computed against the corresponding Control draft. The No-KG ablation (replacing graph-guided exploration with flat top-T search) degrades creativity most strongly in Draft 1 and the Final merge.

DRAFT	CONTROL	NO KG	δ	EFFECT
DRAFT 1	24.45	23.23	0.56	LARGE
DRAFT 2	24.37	23.84	0.28	SMALL
DRAFT 3	24.45	23.81	0.28	SMALL
FINAL MERGE	25.11	23.76	0.52	LARGE

base through its iterative questioning mechanism. However, the Final merge, which synthesizes across all drafts, cannot fully recover the lost diversity, resulting in a large overall effect.

Table 10 presents the single-hop ablation results, showing the effect of decreasing $\mathcal{N}_c$ to k=1 (5 nodes max) from k=3 (30 nodes max).

The single-hop ablation shows the largest effect in Draft 1 ($\delta = 0.68$), suggesting that richer graph context is most beneficial during initial synthesis when the agent must establish its creative framework. The Final merge also shows a large effect ($\delta = 0.52$), confirming that the broader neighborhood context provides lasting benefits.

To further understand why graph structure matters, creativity scores were compared between the top 10 hub nodes (high connectivity) and bottom 10 leaf nodes (low connectivity) for the G that was explored for each challenge. Overall, hub node insights scored significantly higher than leaf insights (24.43 vs. 21.88, $\delta = 1.00$). This indicates that hub nodes, which aggregate information from diverse branches, are particularly valuable for creative synthesis.

Table 10: Single-hop neighborhood ablation by draft iteration (Full Answers). Decreasing $\mathcal{N}_c$ to k=1 causes consistent performance degradation over the standard k=3 configuration.

DRAFT	CONTROL	SINGLE-HOP	δ	EFFECT
DRAFT 1	24.75	23.49	0.68	LARGE
DRAFT 2	24.80	23.71	0.32	SMALL
DRAFT 3	24.88	23.72	0.44	MEDIUM
FINAL MERGE	25.45	23.69	0.52	LARGE

H Compute-Controlled Comparison

To address the confound of uncontrolled compute budgets across baselines, a cost-controlled experiment was conducted. Caesar was limited to T=250 exploration steps using GPT-5-mini for Phase 1, matching the estimated $5 per-challenge cost of Gemini Deep Research. The evaluation follows the same protocol as Table 1 under the unconstrained full-answer setting. As shown in Table 11, Caesar is able to outperform all of the deep research baselines with large values for δ.

Table 11: Cost-controlled comparison. Caesar at T=250 with GPT-5-mini matches the estimated compute budget of Gemini Deep Research while still outperforming all baselines with large effect sizes.

DEEP AGENT	AVG SCORE	δ	EFFECT
CAESAR (T=250, GPT-5-mini)	**26.16**	–	–
GEMINI 3 (DEEP)	24.37	0.76	LARGE
SONNET 4.5 (DEEP)	21.00	1.00	LARGE
GPT-5.2 (DEEP)	16.16	1.00	LARGE

Even under this strict budget constraint, Caesar significantly outperforms all deep research baselines. This result demonstrates that Caesar's architectural advantage (graph-guided exploration + adversarial synthesis) is not merely a function of increased compute expenditure, but reflects genuine algorithmic improvements in how the exploration budget is utilized.

I Human Evaluation Study

To validate the LLM-as-a-Judge framework, a human evaluation study was conducted comparing Caesar against the strongest baseline (Gemini 3 Deep Research, which was rated second best by the LLM judges).

I.1 Study Design

23 human raters were recruited to evaluate pairwise comparisons between Caesar and the baseline. The full answers of both agents were passed through the same LLM-based normalization step (Listing 7) that rewrites each answer into a fixed two-paragraph schema: a 2–3 sentence plain-language summary of the core idea (prefixed `SUMMARY:`) and a 3–4 sentence argument for why it is creative (prefixed `WHY IT'S CREATIVE:`). This format was explicitly designed to allow raters to quickly judge the underlying idea while not getting distracted by surface verbosity or structure. Human judges were then asked to select which of two anonymized normalized answers was more creative according to the NUS rubric used by the LLM judges. Each of the five challenges produced multiple comparison pairs, yielding 112 total A/B matchups.

Listing 7: Human Evaluation Normalization Prompt

```
ARTIFACT: {artifact_text}

YOUR TASK:
Write exactly two short paragraphs about the artifact above.

PARAGRAPH 1 - CORE IDEA SUMMARY (2-3 sentences):
```

```
- Identify and summarize the most important creative key idea in the
    artifact
- Make sure the summary is clear and understandable to a non-expert
- The reader should immediately understand what the artifact is about
- Do not overuse jargon or too many technical terms

PARAGRAPH 2 - CREATIVITY ARGUMENT (3-4 sentences):
- Make a clear, convincing argument for why the key idea is creative
- Explain what makes it novel, surprising, or useful
- Ground your argument in specific aspects of the idea, not generic
    praise
- Write to persuade a skeptical reader

IMPORTANT: Do NOT mention or reference the artifact; write as if
    describing the idea itself.
IMPORTANT: Start paragraph 1 with "SUMMARY: " and paragraph 2 with "
    WHY IT'S CREATIVE: "
```

I.2 Results

Table 12: Human evaluation results: pairwise preference counts for Caesar vs. Gemini 3 Deep Research across all five challenges (C1: Constrained Synthesis, C2: Counterfactual Reasoning, C3: Cross-Domain Synthesis, C4: Meta-Creativity, C5: Open-Ended Synthesis).

MODEL	C1	C2	C3	C4	C5	TOTAL
CAESAR	5	18	20	13	7	**63**
GEMINI 3	18	5	2	9	15	49

Human raters preferred Caesar overall (63 vs. 49, 56.25% preference rate, odds ratio 1.29). The pattern across challenges mirrors the LLM judge results: Caesar is strongly preferred for tasks requiring cross-domain synthesis (C3: 20 vs. 2) and counterfactual reasoning (C2: 18 vs. 5), while the preference is reversed for constrained synthesis (C1: 5 vs. 18) and open-ended synthesis (C5: 7 vs. 15). This alignment between human and LLM judge evaluations supports the reliability of the automated scoring framework.

J Caesar Implementation Details

This section outlines the technical specifications of the Caesar architecture, detailing the software stack used for graph management and web perception. It also provides the set of hyperparameters configured for the exploration and synthesis phases to ensure reproducibility.

J.1 System Architecture

The Caesar agent framework was implemented using the Python programming language. The core components that use external libraries are:

- **Graph Management:** The navigational graph G is managed using `NetworkX` [Hagberg et al., 2008]. Each node represents a visited URL, containing attributes for the raw text content, the timestamp of access, and the depth relative to the root node. The graph is directed, with edges representing navigational transitions (e.g., clicking a link).
- **Vector Store:** The store utilizes `ChromaDB` [Chroma Core, 2024] for the knowledge base *KB*. Text chunks are embedded using `text-embedding-3-large` (via OpenAI) with a chunk size of 400 tokens and an overlap of 80 tokens. The system supports metadata filtering based on iteration and depth.
- **Web Perception:** To mitigate anti-bot measures, a custom `curl_cffi` wrapper (configured to impersonate Chrome) was used to fetch HTML content. This wrapper handles TLS fingerprinting, automatic decompression, and header management to mimic legitimate browser traffic. Content

extraction and filtering of irrelevant text (e.g., SEO links) is handled by `Trafilatura` [Barbaresi, 2021] for HTML and `PyPDF2` [Fenniak et al., 2022] for PDF documents.

- **LLM Backend:** The experiments were conducted using GPT-5.2 [OpenAI, 2025] as the primary driver for both exploration and synthesis. The system uses low reasoning effort during the exploration phase to increase throughput and high reasoning effort during the artifact synthesis phase to ensure quality of answers.
- **Memory Layer:** An exploration memory store was implemented using the `Mem0` library [Singh et al., 2024], which integrates a `ChromaDB` vector store with a `Neo4j` [Neo4j, Inc., 2024] graph database for managing long-term agent memory and detecting navigational loops.

J.2 Hyperparameters

Table 13 details the specific hyperparameters used for the main experiments reported in Section 4.

Table 13: Hyperparameter settings for both web exploration (Section 2) and artifact synthesis (Section 3) phases of Caesar.

Parameter	Description	Value
	Phase 1: Deep Web Exploration	
T	Exploration Budget (Steps)	1000
P_m	Max Page Content (Char)	100k
L_m	Max Candidate Links per Page	2000
R_m	Max Allowed Page Revisits	20
S_m	Max Web Search Actions	30
D_m	Max Graph Exploration Depth	10000
$\lvert\mathcal{N}_c\rvert$	Max Graph Neighbor Context Size (k=3 hops)	30
τ_e	LLM Temperature for Think/Act	0.9
R_e	LLM Reasoning Effort for Think/Act	Low
	Phase 2: Adversarial Artifact Synthesis	
$\hat{T}$	Recursive Insight Budget (Iter)	30
N	Adversarial Refinement Rounds	3
H_c	Max QA Context History	50
C_m	Max Citations per Claim	5
τ_s	LLM Temperature for Draft/Merge	0.1
R_s	LLM Reasoning Effort for Draft/Merge	High
	Global Settings	
O_m	Max LLM Output (Tokens)	50k
R_k	Top-k KB Retrieval	50
R_n	Top-n KB Reranking	10

K Prompts for Caesar

This section shows the LLM instructions from both the exploration and synthesis phases of Caesar. These prompts (abridged for readability) demonstrate how Caesar moves beyond standard RAG retrieval by carefully incorporating information from the knowledge graph and past drafts as context.

K.1 Phase 1 Prompts (Deep Web Exploration)

Graph-Augmented Insight Generation (Think). The LLM evaluates the relationship between the current page content and insights from local knowledge graph nodes. Rather than summarizing the page in isolation, Caesar identifies narrative gaps and unexpected connections using the insights as context.

Listing 8: Graph-Augmented Insight Generation Prompt

```
PAGE CONTENT: {page_content}

INITIAL QUERY: {initial_query}

PAST INSIGHTS: {past_insights}
```

```
NEIGHBOR INSIGHTS: {neighbor_insights}

YOUR TASK:
Analyze this content and extract key insights focusing on:
- Novel patterns or unexpected connections
- Assumptions being made and alternative perspectives
- Interesting questions raised by the content
- How to answer the query
- How this builds upon or challenges past/neighbor insights

Depending on the complexity of the content, provide anywhere from 1 to
    6 concise but substantive insights, but do not exceed ~600 words
   in total length:
```

Knowledge-Guided Exploration (Act). To prevent navigational amnesia, a high-level meta-strategy (EXPLORE, BACKTRACK, or WEBSEARCH) is determined based on the current state of exploration. Afterwards, the next link is selected based on that strategy.

Listing 9: Knowledge-Guided Exploration Prompt

```
CURRENT EXPLORATION CONTEXT:
- Current step: {current_step}/{max_steps}
- Current depth: {current_depth}/{max_depth}
- Web pages visited: {len(visited_urls)}
- Current URL: {current_url}

CURRENT EXPLORATION INSIGHTS:
{kb_context if kb_context else "No exploration insights available."}

HISTORICAL NAVIGATION PATTERNS:
{memory_context if memory_context else "No exploration history
   available."}

Analyze whether the agent should:
1. **EXPLORE** new un-visited pages to discover novel information or
   knowledge
2. **BACKTRACK** to the immediate previously visited page to try
   alternative paths
3. **WEB_SEARCH** relevant topics to address current exploration
   insights

Consider:
- Knowledge gaps vs areas of saturation
- Depth of current exploration branch
- Success patterns from previous decisions
- Risk/reward of new exploration vs consolidation
```

K.2 Phase 2 Prompts (Adversarial Artifact Synthesis)

Recursive Insight Discovery. During the initial insight discovery stage, the agent generates the next logical question in the chain of thought using the previous question/answer insights as context.

Listing 10: Recursive Insight Discovery Prompt

```
PREVIOUS INSIGHTS: {list_of_qa_insights}

YOUR TASK:
Based on the insights gathered so far, what is the next most important
    question to ask
to deepen understanding and reveal emergent patterns? The question
   should:
- Build on previous insights rather than repeat them
- Seek connections between different themes
- Identify gaps or contradictions to explore
```

```
- Move toward synthesis and creation rather than enumeration
```

Recurrent Draft Generation. In the multi-draft loop, this prompt generates the artifact. Crucially, if a previous draft exists, the prompt injects it into the context and explicitly instructs the LLM to critique and improve upon the prior work.

Listing 11: Recurrent Draft Generation Prompt

```
KEY INSIGHTS: {list_of_qa_insights}

PREVIOUS ARTIFACT: {artifact_text}

YOUR TASK:
Drawing heavily upon the patterns that emerged from the key insights,
    and building upon the previous artifact, create a novel, exciting,
     and thought provoking artifact that creatively answers this query
    : {starting_query}
- Emergent patterns not visible in individual sources
- Novel discoveries, connections, or applications
- Surprising new directions or perspectives
- Interesting tensions, contradictions, or open questions

IMPORTANT: do NOT mention or reference the previous artifact, the new
    artifact should make sense by itself as a standalone text.
IMPORTANT: Avoid excessive jargon, ensure artifact text is well-
    organized (logical, clear, focused), and convincing to a skeptical
     reader
```

Adversarial Challenge Refinement. Between drafts, the LLM is prompted to analyze the previous artifact text for weaknesses and formulate a new, refined challenge that targets these weaknesses for the next round of synthesis.

Listing 12: Adversarial Challenge Refinement Prompt

```
PREVIOUS QUERY: {previous_query}

PREVIOUS ARTIFACT: {artifact_text}

YOUR TASK:
Based on the previous query and artifact above, identify the most
    promising direction for deeper exploration. What NEW question or
    angle would:
- Build on the insights already discovered
- Explore gaps, contradictions, or unexplored connections
- Lead to novel perspectives or applications
- Go deeper rather than broader

The refined query should be concise (1-2 sentences), straightforward,
    clear, and understandable.
```

Generative Draft Merging. Finally, the LLM merges all rounds of artifact drafts into a single coherent narrative, identifying tensions and unifying perspectives.

Listing 13: Generative Draft Merging Prompt

```
ARTIFACT DRAFTS: {list_of_artifact_drafts}

YOUR TASK:
Create a comprehensive merged artifact that:
- Combines the draft artifacts into a single cohesive and complete
    artifact
- Selectively integrates the most interesting, relevant insights
    across all draft artifacts
- Discovers emergent patterns not visible in individual artifacts
```

```
- Further develops the core strengths while addressing the weaknesses
    of the draft artifacts
```

Post-Processing and Summarization. The post-processing step distills the final merged artifact A_f into ELI5 language using the prompt shown in Listing 2 (Section B); the same prompt is applied uniformly to all agents during evaluation.

L Computational Cost Analysis

While Caesar achieves higher creativity scores, its heavy usage of LLMs for web page processing and vector store retrieval incurs a noticeable computation cost. The total cost is split between the extensive data gathering in Phase 1 and the recursive draft generation in Phase 2. The computational cost suggests that Caesar is best utilized as a tool for high-value, asynchronous research rather than real-time interaction, similar to the deep research agents described in Li et al. [2025].

L.1 Phase 1: Deep Web Exploration Costs

The exploration phase constitutes the primary computational expense, averaging approximately $40 to $60 per 1000-step experiment. Given the GPT-5.2 pricing of $1.75 per 1M input tokens and an average context load of $\sim$30k tokens per step, this cost reflects the cumulative impact of continuous context usage and output generation during web exploration. The token budget is predominantly consumed by two mechanisms:

- **Ingestion of Text-Heavy Documents:** Caesar processes most or all of the text content of a page to extract deep insights. In most cases, Caesar processes standard web articles with only $\sim$5k tokens. However, when exploring scientific and humanitarian domains, the agent can encounter dense PDF reports and data portals. These text-heavy pages can reach the truncated limit of P_m (100k characters), consuming 20k–25k tokens for the page content alone.
- **Processing and Selecting Links:** The other major consumer of the token budget is the link selection in the "Act" stage. To make informed navigational decisions, the agent processes hundreds of links per page on average (ranging from $\sim$500 on wiki-style pages to 1000+ on dense index pages). Assuming each link consumes $\sim$25 tokens of context, this results in an overall cost of $\sim$10 to 20k tokens per navigation step, regardless of the remaining page content.

L.2 Phase 2: Adversarial Artifact Synthesis Costs

In contrast, the synthesis phase is comparatively inexpensive, averaging less than $3 per round.

- **Efficient Insight Retrieval:** Unlike the exploration phase, which ingests raw uncompressed web content, the synthesis phase operates on a curated knowledge base (*KB*). By limiting the number of insights retrieved and performing reranking on them to generate concise answers, the context window is significantly reduced, minimizing input token costs.
- **High Reasoning Overhead:** The cost driver in this phase shifts to output tokens. The synthesizer utilizes GPT-5.2 with high reasoning effort during the adversarial refinement rounds ($N = 3$). While the input volume is low, the generation of extensive chain-of-thought reasoning [Long, 2023] to integrate disparate insights increases the cost per inference step relative to standard generation.

M Broader Impacts and Responsible Use

The research presented in this paper advances the capabilities of autonomous agents in synthesizing complex information from the open web. However, the architecture introduced in Caesar, specifically the decoupling of graph-based exploration from artifact synthesis, necessitates careful consideration of societal risks.

First, the deployment of recursive web agents raises concerns regarding infrastructure strain and data privacy. While Caesar utilizes browser fingerprinting to ensure robust compatibility with modern web rendering standards during benchmarking, responsible real-world deployment requires

strict adherence to `robots.txt` exclusion protocols and domain-level rate-limiting to preserve site integrity.

Second, the system's objective to maximize creativity introduces a dual-use risk regarding the generation of persuasive misinformation. While the proposed citation mechanism (Section 3) is designed to enforce provenance, the potential for agents to construct compelling but factually tenuous narratives remains an issue. Consequently, this framework is intended as a human-in-the-loop research support tool rather than as a fully autonomous content generator, underscoring the need for continued development of verifiable constraints for creative generation.

In accordance with relevant policies on the use of generative AI, this work utilized LLMs to assist in the generation of Python code for the experimental framework and to improve the clarity and grammatical flow of the manuscript. However, all scientific claims, algorithm/experimental designs, verification of results, and creation of the written content were done by the human authors.