IDEA-EVOLUTION: Dynamic Representations for Long-Horizon, LLM-Supported Research Ideation

Anonymous ACL submission

Abstract

LLMs can generate research ideas at scale, but more ideas do not necessarily lead to better science: automated ideation may elaborate the first plausible proposal while leaving weak assumptions, unresolved limitations, and post-hoc grounding intact. We introduce **IDEA-EVOLUTION**, a framework for representing and analyzing LLM-supported ideation as evolving structured scientific reasoning. IDEA-EVOLUTION constructs and updates idea graphs that capture assumptions, evidence, methods, limitations, and dependencies across proposal revisions. Across five simulated single-agent and multi-agent pipelines and a human study with 15 researchers, our graphs align well with abstracts/proposals (90.9/82.0) and human judgments (85.3). This representation lets us diagnose a central failure of LLM ideation: even across trajectories of up to 20 turns, models often converge prematurely on the initial idea, making only minimal refinements while failing to meaningfully address the critical assumptions and limitations introduced by that idea. Finally, we explore using idea graphs for test-time guidance, improving assumption/limitation resolution by 21% on average and trajectory productivity by 7%, overall suggesting that structured scientific reasoning can help augment LLM and human ideation [1].

1 Introduction

Large language models (LLMs) are rapidly becoming part of the research workflow, shaping how researchers search the literature, generate ideas, design experiments, and write papers (Baek et al., 2024; Gottweis et al., 2025; Lu et al., 2024a; Shao et al., 2024). While these systems make it easier to generate plausible research ideas at scale (Si et al., 2025b; Lu et al., 2024a; Si et al., 2026; Yamada et al., 2025), they also make it harder to determine which ideas are worth pursuing and how to guide

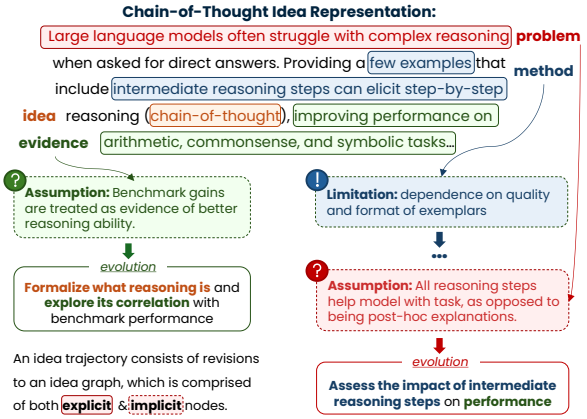

Figure 1: Idea graphs represent research ideas as evolving networks of explicit and implicit nodes (e.g., problems, assumptions, evidence, methods, limitations, etc.).

them toward stronger grounding, quality, and impact (Gupta and Pruthi, 2025; O'Neill et al., 2025; Kargupta et al., 2026).

Much of the current focus on automating research risks exacerbating, rather than addressing, longstanding challenges in scientific discovery. Post-hoc literature reviews may reinforce confirmation bias and weakly grounded claims; implicit assumptions may remain unexamined; and important limitations may go unacknowledged. The central challenge is therefore not simply how to make LLMs generate more ideas, but how to use them to *augment researchers' ability to reason critically and creatively throughout ideation* (Lubart, 2001; Wechsler et al., 2018; Dwyer et al., 2025).

To address this, we tackle a fundamental question: what does "ideation" entail? We frame LLM-supported ideation around three questions. First, *what is an idea?* That is, how can we concretize an abstract concept so it can be analyzed, compared, and improved? Second, *how does an idea evolve?* Modeling ideation as a trajectory, rather than a static output, allows us to track transformations through assumptions, evidence, limitations, and refinements. Third, *how can we diagnose the quality of an ideation trajectory?* To study these ques-

[1] We will release all related code and data upon acceptance.

tions, we introduce **IDEA-EVOLUTION**, a framework that constructs idea graphs from research artifacts and updates them across ideation trajectories, enabling us to analyze how an idea's reasoning evolves over time:

Idea representations should expose explicit and implicit scientific reasoning. We first formalize research ideas through a knowledge-graph-based representation that makes explicit the scientific reasoning often compressed into an abstract or proposal. Drawing on scientific discourse and knowledge representation work (Ciccarese et al., 2008; Liakata et al., 2010; Feng et al., 2025; Kargupta et al., 2025c,b; O'Neill et al., 2025), our schema captures semantic roles and relations involved in ideation: the motivating problem, assumptions, methods, evidence, limitations, and implications. Figure 1 illustrates this for Chain-of-Thought (Wei et al., 2022), including the *problem* of improving LLM reasoning, a *method* based on intermediate reasoning examples, *evidence* from benchmark improvements, *assumptions* about genuine reasoning ability, and *limitations* about exemplar quality and format. The graph also captures how these components relate, revealing fragile parts of an idea, such as claims resting on weak evidence or methods introducing critical limitations.

Idea trajectories should capture conceptual development. Building on this representation, we model ideation as a sequence of updates to an idea graph. Sosa (2019); Gonçalves and Cash (2021) present the ideation lifecycle as an evolving graph, which inspires IDEA-EVOLUTION. Each version of an idea corresponds to a graph capturing the current state of the idea, allowing us to distinguish surface-level edits from substantive conceptual development, such as replacing a method, narrowing an unsupported claim, adding evidence for an assumption, or addressing a limitation (e.g., in Figure 1, formalizing reasoning). By tracking these updates, we can study whether LLM ideation meaningfully develops an idea or simply elaborates around the first plausible proposal.

Graph-derived signals reveal trajectory-level weaknesses. The same representation also exposes analysis signals that are difficult to measure from text alone. If the core method remains unchanged, the trajectory may be converging prematurely; if a claim depends on an ill-supported assumption, the idea may be weakly grounded; and if limitations accumulate without being resolved, the proposal may be becoming longer without becoming conceptually stronger. In this work, we treat these signals primarily as dimensions for analyzing ideation trajectories, while also exploring in a small case study how they can be used as revision targets.

In this work, we move beyond isolated idea evaluation toward a structured account of how ideas are represented, developed, and diagnosed over time:

- We introduce IDEA-EVOLUTION, a training-free framework that represents research ideas as dynamic graphs of explicit and implicit reasoning, including assumptions, limitations, evidence, methods, and resources (validated across multiple domains with 15 human researchers).
- We analyze five ideation pipelines and reveal premature convergence: even over up to 20 turns, LLMs mostly refine the initial idea and often fail to resolve its core assumptions and limitations.
- We explore graph-guided test-time feedback, improving assumption/limitation resolution by 21% and trajectory productivity by 7% on average, highlighting the potential of structured representations for augmenting ideation.

2 Related Works

LLM-supported scientific ideation. Recent work has explored LLMs for generating, evaluating, and refining scientific ideas. Large-scale human studies and benchmarks show that LLMs can produce novel research ideas, but that feasibility, diversity, originality, and execution remain persistent challenges (Si et al., 2025b,a; Gupta and Pruthi, 2025; Wenger and Kenett, 2025). Other systems support ideation through literature-grounded generation, facet recombination, structured paper data, or interdisciplinary inspiration (Baek et al., 2024; Radensky et al., 2024; O'Neill et al., 2025; Kargupta et al., 2026). Unlike these works, IDEA-EVOLUTION focuses on representing and tracing how an idea's scientific reasoning evolves across revisions.

Agentic and multi-agent scientific workflows. A growing line of work studies iterative and agentic workflows for research, including self-refinement, reflection, role-play, debate, and autonomous scientific discovery (Jiang et al., 2024; Yang et al., 2026; Shinn et al., 2023; Lu et al., 2024b; Kargupta et al., 2025a; Lu et al., 2024a; Gottweis et al., 2025; Shao et al., 2024). Multi-agent ideation systems further show that agent roles, critique, and collaboration structure can improve scientific idea generation (Su

Node type	Description
`problem`	A problem, question, goal, or motivation framing what needs attention and why.
`background`	Background knowledge, prior work, or contextual setup needed to understand the idea.
`assumption`	An implicit or explicit premise taken as true without direct support in the artifact.
`idea`	A concept, claim, or hypothesis being proposed or asserted; what is posited rather than how it is implemented.
`method`	A method, procedure, approach, model, or experiment design/setup describing how something is done or tested.
`evidence`	Evidence, observation, or result that provides empirical or argumentative support.
`limitation`	A weakness, downside, or constraint.
`implication`	An implication, conclusion, broader consequence, or significance of the idea.
`resource`	A resource such as a material, dataset, tool, or other asset used by the work.
`other`	A fallback type used only when no other type fits.

Table 1: Node types used to represent the semantic roles of concepts in an idea graph.

et al., 2025). IDEA-EVOLUTION is complementary: rather than only changing the interaction protocol, it exposes the evolving idea state as a graph of assumptions, evidence, methods, limitations, and dependencies.

Scientific discourse and knowledge representations. Prior work has represented scientific claims, discourse, and scholarly knowledge using ontologies, taxonomies, retrieval-augmented claim analysis, and knowledge-graph style structures (Ciccarese et al., 2008; Kargupta et al., 2025c,b). These approaches motivate structured representations of scientific reasoning. IDEA-EVOLUTION adapts this perspective to early-stage ideation by constructing dynamic idea graphs from proposal revisions and using graph-level weaknesses to analyze trajectory quality.

3 Idea Graph

3.1 Preliminaries

Definition 1 (Idea). We define an *idea* as a typed, directed graph $G = (V, E)$. Each node $v \in V$ is an atomic concept in a research artifact (e.g., *problem*, *abstract*, *proposal*), such as a problem, assumption, method, evidence statement, limitation, or implication (Table 1). Each edge $e = (v_i, r, v_j) \in E$ expresses a directed semantic relation r between two concepts; the full edge ontology is provided in Appendix A. This representation treats a research idea as structured scientific reasoning rather than as a unstructured text span.

We note that our ontology is designed to be domain-agnostic, allowing the same representation to cover ideas from machine learning, biology, and other research areas while preserving the role each concept plays in the proposal.

Definition 2 (Idea Node). An idea node is an atomic conceptual unit in a research artifact. Each node is assigned a semantic type from the ontology in Table 1, indicating the role the concept plays in the idea. We represent the concept with both a domain-specific *node text*, which preserves the artifact's original meaning, and a *domain-agnostic text*, which restates the same concept in more general mechanistic language. The domain-agnostic form helps compare ideas across fields and surface analogies that would otherwise be hidden by specialized terminology.

Nodes may also be marked as *implicit* when they capture assumptions, dependencies, constraints, or limitations that are strongly implied but not directly stated. This is important for ideation because many early-stage ideas are limited by unstated premises rather than by their explicit claims. For practical interpretability, we also associate each node with a short label and supporting evidence explaining how the concept is grounded in the artifact. Table 2 shows an illustrative node.

Field	Example value
`Node Type`	`assumption`
`Label`	Reasoning-step faithfulness assumption
`Node Text`	Chain-of-Thought prompting assumes that intermediate steps improve genuine reasoning.
`Domain-Agnostic Text`	Showing intermediate work is assumed to help the system solve the task.
`Implicit?`	`true`
`Evidence`	The proposal links intermediate reasoning steps to better complex reasoning, leaving the causal mechanism implicit.

Table 2: Example node within an idea graph. These fields are stored for each explicit or implicit concept.

Definition 3 (Idea Trajectory). An *idea trajectory* is a sequence of graph states $\mathcal{T} = (G_1, \ldots, G_T)$ induced by successive revisions of a research artifact. Moving from G_t to G_{t+1} consists of graph edit operations over nodes and edges: ***(a)*** adding a newly introduced concept, ***(b)*** deleting a concept that no longer belongs to the proposal, ***(c)*** updating a surviving concept's text, type, implicit status, or evidence, and ***(d)*** adding, deleting, or retyping directed relations among concepts. This lets us distinguish surface-level rewriting from substantive conceptual development, such as replacing a method, addressing a limitation, or making an assumption explicit.

Definition 4 (Proposal Generation). Our primary task assumes a research problem x as in-

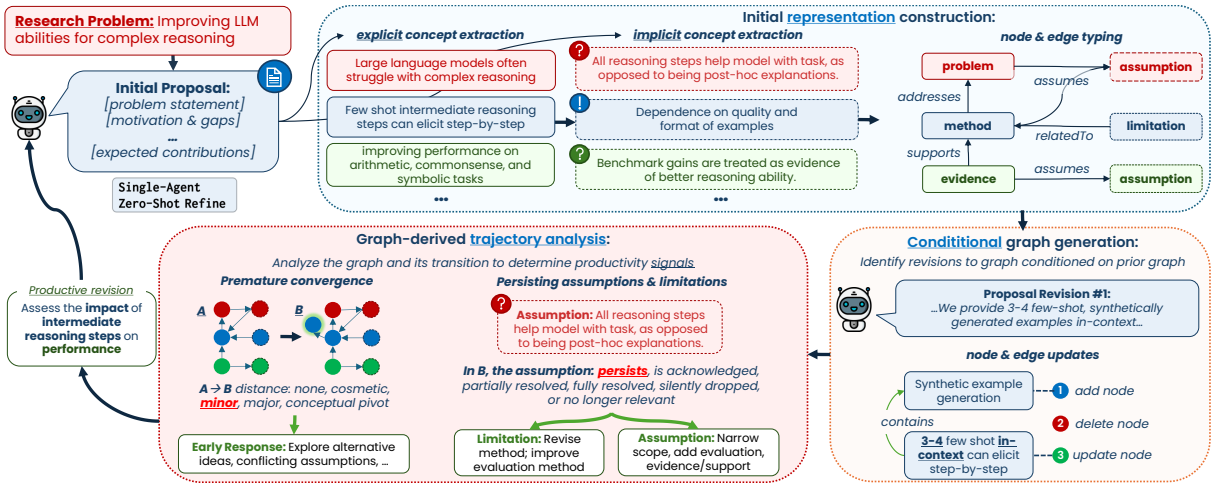

Figure 2: IDEA-EVOLUTION constructs idea graphs, traces graph updates across revisions, and derives graph-level signals for analyzing ideation trajectories. Proposal generation depends on simulation setting (Table 3).

put and produces a sequence of proposal revisions $y_1, \ldots, y_T$. At each step, the current textual artifact y_t is mapped to an idea graph G_t. In our experiments, each proposal revision follows the six-section format in Appendix C. Although we instantiate the task with research problems, the same framework can operate over a variety of textual research artifacts, including problem statements, abstracts, draft proposals, or full papers.

4 IDEA-EVOLUTION

IDEA-EVOLUTION aims to model research ideation as a sequence of structured reasoning states. Given a textual research artifact, or a trajectory of proposal revisions, our goal is to recover not only the final idea but also the conceptual path by which the idea develops. The training-free framework proceeds in three stages. First, it constructs an initial idea graph from the artifact. Second, it updates the graph conditionally as the artifact is revised. Third, it derives trajectory-level analysis signals from the resulting graph sequence, such as premature convergence, unresolved grounding, and persistent limitations.

4.1 Initial Representation Construction

We first construct an idea graph $G_1 = (V, E)$ for a single artifact y_1 (e.g., research problem, abstract, or proposal), where nodes represent conceptual units and edges represent dependencies among them. We decompose this process into four steps:

1. ***Explicit concept extraction.*** The model identifies self-contained concepts directly stated in the artifact, such as problems, methods, evidence, limitations, and implications.
2. ***Implicit concept extraction.*** The model identifies assumptions, dependencies, constraints, or limitations implied by the artifact but not directly stated.
3. ***Node typing.*** Each concept is assigned a semantic role from Table 1.
4. ***Edge typing.*** The model infers sparse directed relations between concepts using the edge ontology in Appendix A.

We note that the resulting graph is not intended to be a complete knowledge graph of all aspects of the artifact; rather, it is a focused representation of the artifact's underlying scientific reasoning. Appendix D provides the prompt templates used for explicit concept extraction, implicit concept extraction, node typing, and edge typing.

4.2 Trajectory-Aware Dynamic Graphs

We next extend the representation to model the artifact's trajectory. Given proposal versions $y_1, \ldots, y_T$, IDEA-EVOLUTION constructs a corresponding graph trajectory $G_1, \ldots, G_T$. Rather than rebuilding every graph independently, each graph G_t for $t > 1$ is generated by updating G_{t-1} conditioned on the initial research problem y_1 (to preserve relevance), previous proposal y_{t-1}, the current proposal y_t, and the prior graph state G_{t-1}. This conditional graph generation is conducted over two stages:

1. ***Node updates.*** For each node type, the model receives the prior nodes of that type and deter-

mines whether each node should be kept, updated, or deleted. It may also add new nodes introduced by the current revision.

2. ***Edge updates.*** After node updates, dangling edges (only connected to a single node) are removed. The model then regenerates the final edge set for each relevant relation type, conditioned on the current nodes and previous graph.

This conditional formulation makes semantic consistency throughout the graphs explicit. A method can persist while its details change; an assumption can become explicit; a limitation can be partially resolved; or a branch of the idea can be abandoned. Thus, the trajectory is represented as a sequence of graph state transitions, enabling structural analysis of ideation at the level of scientific reasoning rather than surface textual change. The node- and edge-update prompts are shown in Appendix D.

4.3 Graph-Derived Trajectory Signals

Finally, we analyze each graph trajectory through signals that summarize how the idea changes over time. Rather than treating the final proposal as the only object of evaluation, IDEA-EVOLUTION compares adjacent graph states G_{t-1} and G_t to characterize the trajectory of the underlying scientific reasoning.

Concretely, a judge compares adjacent graph states and assigns two kinds of feedback: the semantic distance of the revision and the resolution status of assumption and limitation nodes. Low-distance early revisions are treated as signals of *premature convergence* (Wenger and Kenett, 2025; Afzal et al., 2025), while persisting, silently dropped, or newly introduced assumptions and limitations are treated as *unresolved grounding* issues.

These signals turn graph weaknesses into measurable trajectory dimensions. Semantic distance captures whether revisions explore substantially different conceptual territory or repeatedly elaborate the same proposal. Grounding resolution tracks whether assumptions and limitations are acknowledged, partially resolved, fully resolved, silently dropped, or left persistent. We use these dimensions throughout our experiments to compare ideation settings. In addition, Appendix B describes an exploratory case study that uses the same signals as revision targets.

5 Experiments

5.1 Simulating Agentic Ideation

5.1.1 Representation Alignment

We first evaluate whether IDEA-EVOLUTION constructs faithful idea graphs from standalone research artifacts. For static representation alignment, we use 600 arXiv 2025 research abstracts and their corresponding generated research proposals, spanning 17 unique arXiv primary subject areas. This gives us paired source artifacts at two levels of specificity: compact paper abstracts and fuller proposal-style research artifacts.

5.1.2 Trajectory Alignment

For trajectory alignment, we simulate five ideation settings over 51 unique research problems and 20 proposal versions, yielding 5,100 idea states. These settings draw on self-refinement, reflection, and multi-agent scientific workflows (Madaan et al., 2023; Shinn et al., 2023; Lu et al., 2024b; Su et al., 2025; Chen et al., 2025; Si et al., 2025b; Kargupta et al., 2025a, 2026). All settings begin with the same research problem and produce a sequence of proposals, allowing us to compare not only final proposal quality but also how different interaction structures change the trajectory of the idea.

The ***single-agent*** settings vary how much structure the agent receives. In the scaffolded setting, the agent develops the proposal section-by-section, choosing a section, critiquing it, retrieving supporting literature, and revising that section. In the zero-shot setting, the agent first writes a full proposal and then repeatedly performs whole-proposal self-revision without section-level scaffolding. In the free-refine setting, the agent also starts from a full proposal, but chooses which section or weakness to revise at each round.

The ***multi-agent*** settings vary the source of critique. In the domain-scaffolded setting, the input problem x is decomposed into challenges (provided by Kargupta et al. (2026)); for each challenge, domain experts discuss the proposal using domain-specific retrieved literature, and a synthesizer updates the proposal after the discussion. In the domain-agnostic free setting, generic research agents take turns selecting a proposal focus, retrieving relevant evidence, and contributing suggestions or open questions; a synthesizer periodically integrates the discussion into a full-proposal revision. Table 3 summarizes these settings; implementation details for constructing trajectories are provided

Simulation setting	Interaction pattern	Evidence/critique source	Proposal update	Comparison focus
`Single-Agent Scaffolded`	One agent first builds the proposal section-by-section and then iteratively selects the next section to improve.	Section-specific critique and search.	One predetermined section per turn, then chosen after initial draft complete.	Whether localized revision yields clean graph changes.
`Single-Agent Zero-Shot Refine`	One agent writes a full proposal, critiques it, and revises it.	Self-critique plus targeted search.	Full proposal each round.	Whole-proposal self-revision without scaffolding.
`Single-Agent Free Refine`	One agent drafts a proposal, then freely chooses what to revise.	Agent-selected critique and search.	One chosen section per round.	Whether free revision returns to weak components.
`Multi-Agent Domain-Agnostic`	Generic researchers take turns discussing sections, suggestions, and open questions.	Each turn retrieves evidence for chosen section.	Full proposal after periodic synthesis.	Whether open discussion improves proposal development.
`Multi-Agent Domain-Scaffolded`	Each challenge is discussed by domain experts with different views.	Domain-specific retrieved literature.	Proposal updated after each challenge discussion.	Whether structured expert panels add transferable insights.

Table 3: Simulation settings used to generate single-agent and multi-agent ideation trajectories. The settings vary scaffolding, revision granularity, and source of critique.

in Appendix G, with pipeline-specific details in Appendix H.

5.2 Evaluation Metrics

5.2.1 Representation Alignment

We evaluate constructed graphs with `GPT-4.1` and human judges. ***Overall Alignment*** measures artifact-level correspondence between the graph and source text. ***Node Fidelity*** measures whether explicit node types are grounded and correctly typed. ***Latent Validity*** measures whether inferred `assumption` and `limitation` nodes are plausible implicit content. ***Edge Fidelity*** measures whether directed relations are supported. All 1–5 ratings are rescaled to 0–100.

5.2.2 Trajectory Alignment

Trajectory alignment evaluates whether conditional graph updates match the corresponding textual revision. ***Section-Localized Alignment*** maps each changed proposal section to expected node types and scores using `GPT-4.1` whether the graph delta changes the corresponding conceptual region.

5.2.3 Trajectory Analysis

For trajectory analysis, judges operate over constrained graph-delta records rather than raw proposals. ***Productivity*** is the percentage of adjacent proposal pairs for which `SciJudge-30B` (Tong et al., 2026) prefers the later revision; SciJudge is a scientific-taste preference model trained on field- and time-matched high- vs. low-impact paper pairs. All other metrics are assessed with `GPT-5-mini`. ***Trajectory Exploration*** measures whether revisions move into new conceptual territory. ***Section Diversity*** measures whether revisions spread across proposal sections. ***Grounding Resolution*** scores whether assumption and limitation nodes are acknowledged or resolved rather than persisted or silently dropped. We include further details in Appendix E.

6 Results

6.1 Idea Graphs Align with Research Artifacts

We first evaluate whether constructed graphs preserve source-artifact content. Table 5 shows high overall alignment for both abstracts (90.9) and longer proposals (82.0), with especially strong edge fidelity (92.1 and 97.2). Lower background/resource and evidence/implication scores, especially for proposals, mainly reflect recall: judge rationales often note missing fine-grained resources, prior methods, expected findings, or whole aspect categories, rather than incoherent graph structure. Table 9 further suggests domain robustness, with abstract alignment between 89.6–92.2 and proposal alignment between 81.2–82.9 across six arXiv domains. These LLM-judge coverage issues appear minor to human researchers, whose scores remain high across node types.

Human Study. We further evaluated our graph representations with 15 researchers (Table 12) across 37 papers and 10 fine-grained subdomains. Participants evaluated abstract-level graphs through a side-by-side web interface, rating six typed subgraphs on a 1–5 scale. As shown in Table 6, human evaluators rated overall alignment at 85.3, with high scores across node types and higher assumption/limitation scores than the `GPT-4.1` latent-validity judge (79.8 vs. 63.7; 81.0 vs. 68.9). Explicit nodes scored higher on average ($M = 4.55$, $\sigma = 0.61$) than implicit nodes ($M = 4.22$, $\sigma = 0.91$; Mann–Whitney $U = 6518.5$, $p = 0.009$). For the five shared abstracts, all pairwise ratings

Setting / Model	Avg. Turns	Productivity	Trajectory Exploration	Section Diversity	Assumption	Limitation
Single-Agent Zero-Shot Refine						
`GPT-4o-mini`	20.0	84.2	32.5	37.3	24.0	31.0
`Qwen3-8B`	5.4	83.0	16.4	25.0	38.3	39.2
`OSS-120B`	20.0	78.9	53.5	49.3	46.9	51.9
Other single-agent settings						
Free Refine (`GPT-4o-mini`)	20.0	96.2	19.3	39.7	22.0	33.0
Scaffolded (`GPT-4o-mini`)	19.9	97.0	21.0	42.7	32.0	39.0
Multi-agent settings						
Domain-Agnostic (`GPT-4o-mini`)	21.0	90.7	26.5	43.0	25.0	31.0
Domain-Scaffolded (`GPT-4o-mini`)	4.3	93.5	21.3	36.5	22.0	24.0
Domain-Scaffolded (`OSS-120B`)	4.3	90.4	28.2	43.0	36.0	43.0

Table 4: Trajectory-level analysis of proposal revision methods, grouped by agent setting. *Avg. Turns* reports the average number of proposal revision/graph turns per sample; all other metrics are defined in Section 5.2.

Metric	Abstract	Proposal
Overall	90.9	82.0
Problem	61.5	79.3
Background/Resource	63.6	45.3
Idea/Method	81.1	82.8
Assumption	73.9	72.0
Limitation	78.2	81.5
Evidence/Implication	64.7	55.8
Edge	92.1	97.2

Table 5: Representation alignment for abstracts and generated proposals.

Metric	Human	`GPT-4.1`
Overall	85.3	90.8
Problem	87.8	65.0
Background/Resource	89.3	43.3
Idea/Method	91.3	75.0
Assumption	79.8	63.7
Limitation	81.0	68.9
Evidence/Implication	87.3	75.8
Edge	—	91.7

Table 6: Human and LLM judge alignment on a subset of 30 abstracts. Human scores are not reported for edges.

Changed Section	N	Alignment
Problem statement	206	72.6 ± 25.9
Motivation gap	215	59.0 ± 23.0
Method	176	73.0 ± 23.7
Evaluation plan	174	50.7 ± 30.3
Risks / limitations	173	52.7 ± 29.3
Expected contributions	169	53.0 ± 23.6

Table 7: Section-localized alignment between textual updates and graph deltas for single-agent scaffolded trajectories. Values report mean $\pm$ standard deviation across the 20 revisions.

were within one point, suggesting that researchers who know the area largely agree on whether an inferred assumption or limitation is reasonable. Additional study details are provided in Appendix F; the participant-facing protocol and interface are shown in Appendix I.

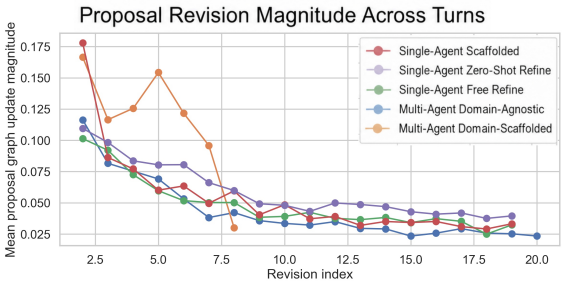

Figure 3: Proposal graph update magnitude over revision steps across simulated ideation settings.

Dynamic graphs capture ideation trajectories. We next evaluate whether graph updates align with proposal revisions. In the scaffolded setting, where the edited section is known, alignment is strongest for problem statements (72.6) and methods (73.0), and weaker for motivation, evaluation, risks, and contributions (Table 7). These values should be read relative to the 1–5 judge scale: scores around 0.6 after rescaling (3/5) still indicate partial alignment, meaning the graph captures the main direction of the textual revision even if it misses certain details or introduces some unsupported text (slightly biased against implicit nodes). This mirrors the static results: concrete problem/method edits are easier to localize, while support, risk, and contribution edits span multiple graph regions.

6.2 LLM Trajectories Show Limited Exploration

Table 4 shows productive revision but limited conceptual search. Later proposals are usually preferred by SciJudge (78.9–97.0), yet exploration is modest: after rescaling, a score near 25 means the model mostly improves the same idea, while a score near 50 means it tries a few related branches and folds them back into the proposal. Most settings therefore improve the initial direction rather than search broadly over alternatives.

Model scale helps unevenly. In zero-shot re-

Steering case study	Product.	+ Signals	Trajectory Exp.	+ Signals	Section Div.	+ Signals	Assump.	+ Signals	Limit.	+ Signals
Zero-Shot (`GPT-4o-mini`)	84.2	92.9	32.5	26.8	37.3	34.4	24.0	48.0	31.0	48.1
Zero-Shot (`Qwen3-8B`)	83.0	89.5	16.4	18.5	25.0	24.7	38.3	41.0	39.2	46.0
Zero-Shot (`OSS-120B`)	78.9	87.2	53.5	46.8	49.3	46.8	46.9	48.0	51.9	50.0
Domain-Scaffolded (`OSS-120B`)	90.4	90.6	28.2	29.5	43.0	38.2	36.0	45.0	43.0	49.0

Table 8: Exploratory signal-guided steering case study across 10 turns (non-steering across 20 turns). Green/red cells indicate improvements/declines relative to the corresponding baseline.

Domain	Abstract	Proposal	Abstract Edge	Prop. Edge
q-bio	89.6	81.2	91.8	97.0
cs	89.9	82.2	94.7	96.5
math	89.9	82.5	89.1	96.9
econ	91.8	82.9	93.7	97.3
physics	92.0	82.1	93.0	98.1
eess	92.2	81.2	90.5	97.6

Table 9: Domain-specific representation alignment and edge-fidelity scores. Each domain contains approximately 100 paper abstracts.

finement, OSS-120B improves exploration (53.5) and section diversity (49.3), but has lower productivity than smaller models and only moderate grounding resolution. Interaction structure also matters: scaffolded and free-refine settings are highly productive (97.0/96.2) but minimally exploratory, while domain-scaffolded multi-agent trajectories achieve nontrivial exploration and grounding despite only 4.3 turns on average. Figure 3 suggests why: they make large early graph updates before tapering. Grounding remains difficult overall: assumption/limitation scores mostly fall between acknowledgment and partial resolution, showing that LLMs often gesture at fragile nodes without repairing them.

Signal-guided case study. Table 8 converts graph signals into lightweight critique targets. Productivity improves in all four settings, and the clearest gains are on the dimensions explicitly targeted by the signals: assumption resolution rises by 1.1–24.0 points and limitation resolution rises in three of four settings. This suggests that naming the fragile assumption or limitation gives the model a more actionable target than a generic request to improve the proposal. However, exploration and section diversity do not consistently improve; we note that this is likely due to comparing 20 vs. 10-turn length trajectories (for cost efficiency).

Table 10 illustrates why the graph representation is useful beyond summarization: the signal does not merely say that the proposal is weak, but names a specific hidden dependency (cache similarity) and asks for a testable revision. The final proposal improves by operationalizing several such dependencies as diagnostics, while still preserving unresolved costs that a researcher would need to inspect next.

Stage	Qualitative evolution in signal-guided trajectory
Problem	The input asks for methods that train high-resolution CNNs for 3D medical imaging or video segmentation under severe GPU-memory limits, where storing activations for back-propagation exceeds hardware capacity.
Proposal #1	The first proposal introduces an adaptive activation-compression scheduler: per-layer activation autoencoders, a memory-aware policy, and LLM-guided hyperparameter prompts. The graph surfaces assumptions about reliable LLM prompts, accurate real-time GPU memory measurement, and cacheable activation similarity; it also flags compression overhead, RL instability, and system complexity.
Graph signal	One critique states: "The cache effectiveness presumes that stored representations share enough features with upcoming data to aid accurate rebuilding." The graph therefore asks the reviser to make the cache assumption explicit or add an evaluation that directly tests it, rather than merely claiming lower memory use.
Proposal #10	The final proposal, *AdaMem3D*, adds NAS-based low-activation backbones, dynamic sparsity, optional model-parallel offloading, cache hit-rate and reconstruction-error diagnostics, KL monitoring for the controller, and gradient-fidelity metrics for compression. Several early assumptions become testable mechanisms, while the graph still exposes unresolved costs such as tiling overhead, offloading latency, and NAS pre-search expense.

Table 10: Qualitative example from a signal-guided domain-scaffolded trajectory with a 100% productivity.

7 Conclusion

IDEA-EVOLUTION reframes LLM-supported research ideation as an evolving graph of scientific reasoning. The graphs align well with abstracts and proposals (90.9 and 82.0 overall), remain robust across six arXiv domains, and are rated highly by human researchers (85.3). More importantly, the trajectory analysis exposes a core weakness of current LLM ideation: revisions often look productive locally (78.9–97.0 adjacent preference) while staying close to the initial idea, mostly refining rather than exploring alternatives. Even larger models and multi-agent protocols do not reliably repair the assumptions and limitations that make proposals fragile. Thus, the problem is not only idea generation quality, but the quality of the path by which ideas develop. Our signal-guided case study suggests that graph-derived weaknesses can become useful intervention targets, but robust conceptual steering remains an open future direction.

8 Limitations

Our experiments use simulated LLM ideation trajectories rather than full human scientific collaborations. This enables controlled comparisons across interaction structures, but real teams may revise ideas with different goals, social dynamics, and domain knowledge. We partially address this by evaluating graphs with researchers across domains, though the study is modest in size and focuses on abstract-level inspection.

IDEA-EVOLUTION also relies on LLM extraction and judging. Human and LLM evaluations suggest the graphs are generally faithful, but dense background, evidence, and implication content remain harder to capture. We therefore report representation alignment, trajectory alignment, and human evaluation separately, and treat trajectory scores as diagnostic signals rather than replacements for expert review. Productivity uses SciJudge preferences as a scalable proxy for perceived impact (while acknowledging that it is trained on citation data and hence may inherent related biases), but may not fully capture feasibility, ethics, or long-term value.

Finally, the signal-guided study is exploratory. It shows that graph-derived weaknesses can guide critiques, especially around assumptions and limitations, but simple critique injection does not always produce deeper conceptual revision. Future work should study tighter human-in-the-loop and model-in-the-loop mechanisms for revising the underlying idea graph itself.

References

Osama Mohammed Afzal, Preslav Nakov, Tom Hope, and Iryna Gurevych. 2025. Beyond" not novel enough": Enriching scholarly critique with llm-assisted feedback. *arXiv preprint arXiv:2508.10795*.

Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. 2024. Researchagent: Iterative research idea generation over scientific literature with large language models. *ArXiv*, abs/2404.07738.

Nuo Chen, Yicheng Tong, Jiaying Wu, Minh Duc Duong, Qian Wang, Qingyun Zou, Bryan Hooi, and Bingsheng He. 2025. Beyond brainstorming: What drives high-quality scientific ideas? lessons from multi-agent collaboration. *arXiv preprint arXiv:2508.04575*.

Paolo Ciccarese, Elizabeth Wu, June Kinoshita, Gwendolyn T. Wong, Marco Ocana, Alan Ruttenberg, and Tim Clark. 2008. The SWAN scientific discourse ontology. *Journal of Biomedical Informatics*, 41(5):739–751.

Christopher P Dwyer, Deaglán Campbell, and Niall Seery. 2025. An evaluation of the relationship between critical thinking and creative thinking: Complementary metacognitive processes or strange bedfellows? *Journal of Intelligence*, 13(2):23.

Tao Feng, Yihang Sun, and Jiaxuan You. 2025. Graph-eval: A lightweight graph-based llm framework for idea evaluation. *arXiv preprint arXiv:2503.12600*.

Milene Gonçalves and Philip Cash. 2021. The life cycle of creative ideas: Towards a dual-process theory of ideation. *Design Studies*, 72:100988.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, and 1 others. 2025. Towards an ai co-scientist. *arXiv preprint arXiv:2502.18864*.

Tarun Gupta and Danish Pruthi. 2025. All that glitters is not novel: Plagiarism in ai generated research. *arXiv preprint arXiv:2502.16487*.

Yucheng Jiang, Yijia Shao, Dekun Ma, Sina Semnani, and Monica Lam. 2024. Into the unknown unknowns: Engaged human learning through participation in language model agent conversations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9917–9955, Miami, Florida, USA. Association for Computational Linguistics.

Priyanka Kargupta, Ishika Agarwal, Tal August, and Jiawei Han. 2025a. Tree-of-debate: Multi-persona debate trees elicit critical thinking for scientific comparative analysis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29378–29403.

Priyanka Kargupta, Shuhaib Mehri, Dilek Hakkani-Tur, and Jiawei Han. 2026. Sparking scientific creativity via llm-driven interdisciplinary inspiration. *arXiv preprint arXiv:2603.12226*.

Priyanka Kargupta, Runchu Tian, and Jiawei Han. 2025b. Beyond true or false: Retrieval-augmented hierarchical analysis of nuanced claims. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29664–29679.

Priyanka Kargupta, Nan Zhang, Yunyi Zhang, Rui Zhang, Prasenjit Mitra, and Jiawei Han. 2025c. Taxoadapt: Aligning llm-based multidimensional taxonomy construction to evolving research corpora. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29834–29850.

Maria Liakata, Simone Teufel, Advaith Siddharthan, and Colin Batchelor. 2010. Corpora for the conceptualisation and zoning of scientific papers. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024a. The ai scientist: Towards fully automated open-ended scientific discovery. *ArXiv*, abs/2408.06292.

Li-Chun Lu, Shou-Jen Chen, Tsung-Min Pai, Chan-Hung Yu, Hung-yi Lee, and Shao-Hua Sun. 2024b. Llm discussion: Enhancing the creativity of large language models via discussion framework and role-play. *arXiv preprint arXiv:2405.06373*.

Todd I Lubart. 2001. Models of the creative process: Past, present and future. *Creativity research journal*, 13(3-4):295–308.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *ArXiv*, abs/2303.17651.

Charles O'Neill, Tirthankar Ghosal, Roberta Răileanu, Mike Walmsley, Thang Bui, Kevin Schawinski, and Ioana Ciucă. 2025. Sparks of science: Hypothesis generation using structured paper data. *arXiv preprint arXiv:2504.12976*.

Marissa Radensky, Simra Shahid, Raymond Fok, Pao Siangliulue, Tom Hope, and Daniel S. Weld. 2024. Human-llm compound system for scientific ideation through facet recombination and novelty evaluation.

Yijia Shao, Yucheng Jiang, Theodore Kanell, Peter Xu, Omar Khattab, and Monica Lam. 2024. Assisting in writing Wikipedia-like articles from scratch with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6252–6278, Mexico City, Mexico. Association for Computational Linguistics.

Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems 36*.

Chenglei Si, Tatsunori Hashimoto, and Diyi Yang. 2025a. The ideation-execution gap: Execution outcomes of llm-generated versus human research ideas. *arXiv preprint arXiv:2506.20803*.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. 2025b. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers. In *International Conference on Learning Representations*, volume 2025, pages 94003–94092.

Chenglei Si, Zitong Yang, Yejin Choi, Emmanuel Candès, Diyi Yang, and Tatsunori Hashimoto. 2026. Towards execution-grounded automated ai research. *arXiv preprint arXiv:2601.14525*.

Ricardo Sosa. 2019. Accretion theory of ideation: Evaluation regimes for ideation stages. *Design Science*, 5:e23.

Haoyang Su, Renqi Chen, Shixiang Tang, Zhenfei Yin, Xinzhe Zheng, Jinzhe Li, Biqing Qi, Qi Wu, Hui Li, Wanli Ouyang, Philip Torr, Bowen Zhou, and Nanqing Dong. 2025. Many heads are better than one: Improved scientific idea generation by a LLM-based multi-agent system. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28201–28240, Vienna, Austria. Association for Computational Linguistics.

Jingqi Tong, Mingzhe Li, Hangcheng Li, Yongzhuo Yang, Yurong Mou, Weijie Ma, Zhiheng Xi, Hongji Chen, Xiaoran Liu, Qinyuan Cheng, and 1 others. 2026. Ai can learn scientific taste. *arXiv preprint arXiv:2603.14473*.

Solange Muglia Wechsler, Carlos Saiz, Silvia F Rivas, Claudete Maria Medeiros Vendramini, Leandro S Almeida, Maria Celia Mundim, and Amanda Franco. 2018. Creative and critical thinking: Independent or overlapping components? *Thinking skills and creativity*, 27:114–122.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Emily Wenger and Yoed Kenett. 2025. We're different, we're the same: Creative homogeneity across llms. *arXiv preprint arXiv:2501.19361*.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. 2025. The ai scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint arXiv:2504.08066*.

Ruozhen Yang, Yucheng Jiang, Yueqi Jiang, Priyanka Kargupta, Yunyi Zhang, and Jiawei Han. 2026. Grounding agent memory in contextual intent. *Preprint*, arXiv:2601.10702.

A Edge Ontology

Idea-graph edges encode directed semantic relations between concept nodes. We use the ontology in Table 11 to distinguish structural, evidential, causal, contrastive, and fallback relations while keeping graphs sparse and interpretable.

Edge type	Description
`contains`	A broader A includes a narrower component, B.
`refines`	A is derived from, evolved from, or refines B.
`motivatedBy`	A exists because of a problem or rationale, B.
`supports`	A provides support for B (e.g., cited evidence).
`refutes`	A contradicts or weakens B.
`alternativeTo`	A is presented as an alternative to B.
`assumes`	A depends on an assumption B.
`addresses`	A method or idea A targets a problem B.
`answers`	A piece of evidence or implication A answers a problem framed as a question B.
`uses`	Method or evidence A uses idea, resource, or method B.
`tests`	Evidence-producing method A tests idea or method B.
`produces`	Method A produces evidence or downstream output B.
`implies`	Evidence, an idea, or an implication A suggests B.
`contrastsWith`	A is explicitly contrasted against B.
`relatedTo`	A weak fallback relation used only if a stronger edge type does not apply.

Table 11: Edge types used to represent directed dependencies and relations among concepts in an idea graph.

B Signal-Guided Case Study Details

The signal-guided case study uses the same graph construction and trajectory judges as the main analysis, but feeds selected signals back into the next revision prompt. For each unresolved assumption node, the action planner asks the reviser to make the premise explicit, add a targeted evaluation or ablation, or add concise support when the assumption is already well integrated with the proposal. For limitation nodes, the planner either revises a connected `idea` or `method` node so the limitation becomes less severe, or adds an evaluation method that measures whether the limitation has been reduced. Newly introduced assumptions and limitations are handled as active issues in the next revision rather than only as diagnostic labels. Early low-distance revisions trigger an alternative-generation action that preserves the original research problem while exploring a different idea, method, or inverted premise. The final action set is selected with a balanced budget over assumptions, limitations, and early-convergence signals. We include this as an exploratory use of the analysis signals.

C Proposal Format

All simulated ideation settings generate and revise proposals using the same structured format. The format is intentionally lightweight: it gives agents enough scaffolding to make comparable revisions, while leaving the scientific content open-ended.

Proposal Revision Format

Task. Given a research problem, generate or revise a complete research proposal. Fill each field directly with concrete proposal content; do not return a prose wrapper.
Fields.

- `problem_statement`: the research problem, why it matters, and the setting in which it arises.
- `motivation_gap`: why existing approaches are insufficient and what conceptual or empirical gap motivates the proposal.
- `method`: the proposed approach, mechanism, intervention, model, or experimental design.
- `evaluation_plan`: datasets, baselines, metrics, ablations, or analyses used to test the proposal.
- `risks_limitations`: assumptions, validity threats, failure modes, feasibility constraints, or ethical concerns.
- `expected_contributions`: the intended scientific contributions and broader implications.

Output schema.

```
{"problem_statement": "...",
"motivation_gap": "...",
"method": "...",
"evaluation_plan": "...",
"risks_limitations": "...",
"expected_contributions": "...",
"change_summary": "..."}
```

D Prompt Templates

We reproduce the main control prompts used by IDEA-EVOLUTION. For readability, we omit long few-shot examples from the extraction prompt and replace instance-specific payloads with bracketed placeholders.

D.1 Initial Representation Prompts

Explicit Concept Extraction

Role. You are extracting concept nodes from a research idea excerpt.
Task. Identify the atomic concepts needed to represent the excerpt as an idea graph.
Instructions.

- A concept node may be a self-contained proposition, claim, observation, method description, assumption, or viewpoint-like unit.
- Split sentences containing multiple separable ideas; replace pronouns, expand acronyms, and rewrite lightly so each concept can stand alone.
- Extract only concepts grounded in the excerpt. Prefer proposition-like concepts over bare keywords.
- Provide `label`, `text`, `domain_agnostic_text`, `node_type`, and `evidence`.
- Use the ontology in Table 1; prefer precision over recall and avoid duplicate concepts.

Output. Return valid JSON only:

```
{"concepts": [{"label", "text",
"domain_agnostic_text",
"node_type",
"evidence"}]}
```

Input. `Excerpt: [source artifact or proposal revision]`.

Implicit Concept Extraction

Role. You are identifying implicit concepts implied by a research idea excerpt.
Task. Given the excerpt and already extracted explicit concepts, identify important implicit concepts that are implied but not directly stated.
Priorities. Assumptions, unstated dependencies, hidden constraints, implicit goals or motivations, implied evaluation criteria, tacit problem framings, and implicit limitations.
Instructions.

- Include only implicit concepts strongly supported by the excerpt.
- Do not repeat explicit concepts unless the implicit node captures a genuinely implicit dependency or framing.
- Most implicit concepts should be typed as `assumption`, `problem`, `limitation`, or `implication`.
- Mark all returned concepts as `implicit=true`; provide confidence and evidence tied to the excerpt.

Output. Return valid JSON only:

```
{"concepts": [{"label", "text",
"domain_agnostic_text",
"node_type",
"implicit": true, "confidence",
"evidence"}]}
```

Inputs. `Existing explicit concepts: [concept JSON]`; `Excerpt: [source artifact]`.

Node Typing

Role. You are assigning node types to concept nodes in an idea graph.
Task. For each concept node, assign exactly one best domain-agnostic node type.
Instructions.

- Assign based on the role the statement plays in the idea, not surface syntax.
- Prefer the most semantically specific type; use `other` only as a last resort.
- Keep `idea` distinct from `method`: `idea` describes what is proposed or asserted, while `method` describes how it is implemented or tested.
- Use the merged ontology in Table 1 consistently.

Output. Return valid JSON only:

```
{"node_type_assignments": [
  {"node_id", "node_type",
  "confidence", "rationale"}
]}
```

Inputs. `Concepts: [concept JSON]`; `Excerpt: [source artifact]`.

Edge Typing

Role. You are assigning typed directed edges between concept nodes in an idea graph.
Task. Given the excerpt and concept nodes, infer meaningful directed relationships between node pairs.
Instructions.

- Output only edges strongly justified by the excerpt or by tightly supported inference.
- Prefer sparse, high-precision structure; do not create fully connected graphs.
- Use the edge ontology in Table 11; use `relatedTo` only when a meaningful relation exists but no stronger type fits.
- Return at most 24 edges, prioritizing central and strongly supported relations.

Output. Return valid JSON only:

```
{"edges": [{"source_id",
"target_id",
"edge_type", "confidence",
"evidence"}]}
```

Inputs. `Concept nodes: [typed node JSON]`; `Excerpt: [source artifact]`.

D.2 Conditional Graph Update Prompts

Conditional Node Update

Role. You are updating only the `[node_type]` nodes of a proposal graph from one revision to the next.
Task. Given the research problem, previous proposal, current proposal, and prior graph nodes of this type, decide which prior nodes should be kept, deleted, or updated, and whether new nodes should be added.
Action rules.

- Use `keep` when the node still fits with no material change.

- Use `update` when the concept persists but its text, type, implicit flag, confidence, evidence, or metadata should change.
- Use `delete` only when the concept is removed, contradicted, superseded, made irrelevant, or fully replaced by another surviving node.
- Be conservative: prefer keep/update over delete unless the concept truly disappears.

Implicit continuity. Implicit nodes are not expected to be restated verbatim. Keep implicit assumptions or limitations when the current proposal still relies on them or remains compatible with them. Delete a implicit assumption only when the revision makes it irrelevant, contradicts it, or adds enough evidence or mechanism that it is no longer an assumption.
Output. Return valid JSON only:

```
{"updates": [
  {"prior_node_id", "action",
"label",
  "text", "domain_agnostic_text",
  "node_type", "implicit",
"confidence",
  "evidence"}
],
"new_nodes": [{. . . }]}
```

Inputs. `Research problem`, `Previous proposal`, `Current proposal`, and `Prior [node_type] nodes`.

Conditional Edge Update

Role. You are updating only the `[edge_type]` edges of a proposal graph from one revision to the next.
Task. Given current graph nodes and prior edges of this type, produce the complete final set of `[edge_type]` edges that should exist in the current revision.
Instructions.

- Return the complete final edge set, not edit operations.
- Only connect node ids that exist in the provided current nodes list.
- Respect allowed source and target node-type constraints for this edge type.
- Prefer sparse, high-precision edges; omit weak or redundant edges.

Output. Return valid JSON only:

```
{"edges": [{"source_id",
"target_id",
"confidence", "evidence",
"metadata"}]}
```

Inputs. `Research problem`, `Previous proposal`, `Current proposal`, `Current graph nodes`, and `Prior [edge_type] edges`.

D.3 Trajectory Signal and Case-Study Prompts

Transition and Weakness Judge

Role. You are a precise research-trajectory judge. Return only JSON. Do not over-credit cosmetic edits as semantic progress.
Task. Judge an online proposal revision and grounding-node resolution.
Scales.

- Semantic distance: `none`, `cosmetic`, `minor`, `major`, `conceptual_pivot`.
- Grounding resolution: `persisted`, `acknowledged`, `partially_resolved`, `fully_resolved`, `silently_dropped`, `no_longer_relevant`.

Inputs. Previous proposal, current proposal, prior assumption and limitation nodes, whether each prior node appears in the current graph, newly introduced assumption and limitation nodes, and directly connected graph nodes.
Output. Return JSON with:

```
{"semantic_distance",
"semantic_rationale",
"grounding_statuses": [
  {"node_id", "status",
"rationale",
  "connected_node_ids_to_address"}
],
"new_grounding_nodes": [
  {"node_id", "node_type", "text",
  "directly_connected_nodes"}
]}
```

Signal-Guided Action Planning

Role. You are converting graph-judge findings into compact revision actions.
Task. Given the transition judgment, select a small balanced set of graph actions for the next revision.
Action space.

- Assumptions: clarify the premise, add a targeted evaluation, or add concise support.
- Limitations: revise a connected idea or method, or add an evaluation method that measures severity reduction.
- Early convergence: explore an alternative idea or method, including one that inverts a current assumption.

Output. Return JSON with:

```
{"actions": [{"action_type",
"severity",
"target_node_id",
"target_node_type",
"target_text",
"connected_node_ids",
"graph_operations",
"proposal_instruction",
"rationale"}]}
```

Signal-Guided Critique Injection

Role. The next critique or revision call receives a graph-level signal plan in addition to the original research problem.
Instruction. Treat the following as a graph edit specification before revising prose. For assumptions, make the premise explicit, add a concrete test, or add concise support only when it fits naturally. For limitations, revise the connected idea or method, or add an evaluation method that measures whether the limitation has been reduced. For early convergence, explore an alternative idea or method before updating the proposal. Keep the revision compact and do not mention graph instrumentation in the final proposal.
Injected action format. Each action contains:

```
{"action_type", "severity",
"target_node_id",
"target_node_type",
"target_text",
"connected_node_ids",
"graph_operations",
"proposal_instruction",
"rationale"}
```

Expected behavior. The model should revise the proposal so that the next conditional graph state can realize the planned graph operations rather than merely adding prose caveats.

E Evaluation Metric Computation

This section describes how the trajectory evaluation numbers in Table 4 are computed from the generated graph artifacts. The implementation first constructs deterministic records from conditional graph trajectories, then applies lightweight semantic judges only to compact records derived from those artifacts. We conducted all open-source analysis on a single NVIDIA H200 GPU, where each graph took approximately 2.5 minutes to construct.

Graph-delta records. For each trajectory, we have a sequence of proposal versions $y_1, \ldots, y_T$ and corresponding idea graphs $G_1, \ldots, G_T$. We skip the initial graph-construction step and evaluate adjacent revision pairs (G_{t-1}, G_t) for $t > 1$. Nodes are matched by `node_id`; edges are matched by the tuple (`source_id`, `target_id`, `edge_type`). A node is marked as updated if any of `label`, `text`, `domain_agnostic_text`, `node_type`, `latent`, or `evidence` changes. An edge is marked as updated if its `edge_type` or `evidence` changes.

Let A_N, D_N, and U_N denote the numbers of added, deleted, and updated nodes, respectively, and let A_E, D_E, and U_E denote the corresponding counts for edges. The total number of changed nodes is $C_N = A_N + D_N + U_N$, and the total number of changed edges is $C_E = A_E + D_E + U_E$. Let $|V_{t-1}|$ and $|V_t|$ be the node counts before and after the revision, and let $|E_{t-1}|$ and $|E_t|$ be the corresponding edge counts. The proposal-level graph update magnitude is

$$M_G = \min(1, R_G),$$
$$R_G = \frac{C_N + 0.5 C_E}{|V_{t-1}| + |V_t| + 0.5(|E_{t-1}| + |E_t|)}.$$

Edges receive half weight because node changes correspond to direct conceptual content changes, while edge changes often refine dependencies among already present concepts.

Section-level graph magnitude. For section-level analyses, proposal sections are mapped to graph node types:

- `problem_statement`: `problem`
- `motivation_gap`: `problem`, `background`
- `method`: `idea`, `method`, `resource`
- `evaluation_plan`: `method`, `evidence`
- `risks_limitations`: `assumption`, `limitation`
- `expected_contributions`: `implication`

For a section bucket s, the section graph magnitude uses only changed nodes whose type belongs to s:

$$\mathrm{Mag}_s(G_{t-1}, G_t) = \min\left(1, \frac{C_{N,s}}{N_{t-1,s} + N_{t,s}}\right).$$

Node lifecycles and branches. For every node id, we record the first and last revision in which it appears, whether it is active in the final graph, its update count, type, section bucket, and text. We also record all node versions for diversity judgments. Deleted `problem`, `idea`, and `method` nodes are candidates for abandoned branches; if they participate in `refines` or `alternativeTo` edges, the branch-edge counts are retained as context for the judge.

LLM trajectory judges. The LLM judges receive compact JSON records rather than raw full trajectories. All judges are instructed to use only the supplied proposal snapshots, node lifecycles, and graph deltas.

- **Trajectory exploration** groups lifecycle records by trajectory, section bucket, and node type for `problem`/`idea`/`method` nodes. The judge sees

when nodes were introduced, updated, dropped, retained in the final graph, and whether any abandoned branch candidates are similar to surviving final nodes. It returns a 1–5 exploration score, where 1 means no temporal exploration, 2 mostly refinement, 3 branching with consolidation, 4 multiple alternatives, and 5 a conceptual pivot.

- **Section diversity** uses the same lifecycle records, grouped by section bucket and node type, and asks whether the set represents a single direction, refinements of one direction, several related directions, or distinct alternatives. Prior-work-only or extraction-noisy nodes are explicitly excluded. Judgeable 1–5 scores are rescaled to 0–100 as $25(\text{score} - 1)$.

- **Grounding resolution** is computed for every `assumption` and `limitation` lifecycle. The judge sees the node lifecycle, the first/last/final proposal snapshots, and deltas for the target node. It assigns one of `persisted`, `acknowledged`, `partially_resolved`, `fully_resolved`, `silently_dropped`, or `no_longer_relevant`. To avoid over-crediting generic mitigation text, `partially_resolved` and `fully_resolved` are accepted only when the judge also identifies a direct intervention or test for the exact issue; otherwise the status is demoted to `acknowledged` or `persisted`.

Scores reported in Table 4. For trajectory exploration and section diversity, a valid 1–5 judge score is converted to a 0–100 score by

$$\text{Score}_{0\text{–}100} = 25(\text{Score}_{1\text{–}5} - 1).$$

For grounding resolution, statuses are mapped to scalar values: `fully_resolved`=100, `partially_resolved`=50, `acknowledged`=25, and `persisted` or `silently_dropped`=0, matching the scoring rule used in Table 4. **Resolution** is the mean over assumption and limitation nodes; **Assump.** and **Limit.** are the same means restricted to the corresponding node type.

F Human Study Details

We evaluated our graph representations with 15 researchers across multiple domains (Table 12). Each researcher submitted two to three abstracts from research proposals or papers they felt confident evaluating, resulting in 37 papers across four broad domains and 10 fine-grained subdomains, including machine learning, computer vision, natural language processing, human-computer interaction, bioinformatics, and medicine. We generated idea graphs from these abstracts using IDEA-EVOLUTION and asked participants to evaluate the resulting representations through a custom web interface displaying the abstract and graph side by side. Evaluations were structured subgraph by subgraph across six typed sections, with participants rating alignment on a 1–5 Likert scale and optionally providing free-text comments. Participants evaluated each graph using only the abstract rather than the full paper.

As shown in Table 6, human evaluators rated overall alignment at 85.3 (rescaled to 0–100), with consistently high scores across node types. For implicit content, human ratings were higher than the `GPT-4.1` latent-validity judge on assumptions (79.8 vs. 63.7) and limitations (81.0 vs. 68.9), suggesting that human evaluators found the inferred content largely reasonable and grounded. Explicit node types scored higher on average ($M = 4.55$, $\sigma = 0.61$) than implicit node types ($M = 4.22$, $\sigma = 0.91$; Mann–Whitney $U = 6518.5$, $p = 0.009$). To assess inter-rater reliability, three researchers independently evaluated the same five abstracts. All pairwise ratings fell within one point on the 1–5 scale (100% within-1 agreement), with idea/method and limitation nodes showing the highest exact agreement (84.6%).

G Trajectory Construction Setup

We generate all proposal trajectories with a shared DSPy-based implementation. Each generator is specified as a set of structured modules for proposal generation, critique, section selection, revision, agent utterance, or synthesis, depending on the simulation setting. All trajectories use the same six-section proposal schema and are saved as turn traces plus edit histories, including the previous proposal, revised proposal, edited target, revision goal, retrieved evidence when applicable, and change summary.

Unless otherwise stated, generation uses OpenAI `gpt-4o-mini` with temperature 0.1 and a maximum output length of 4000 tokens. Full-proposal modes use an AI-Researcher-style initializer that retrieves grounding papers, generates a seed idea, and expands it into the proposal schema.

Retrieval-augmented turns use Paperclip as the scholarly retrieval backend. For each critique find-

ing, selected section, or agent evidence focus, the pipeline generates scholarly queries, retrieves relevant papers, ranks and deduplicates results, and summarizes the selected evidence before revision. We keep the retrieval backend fixed across settings so that differences in trajectories primarily reflect the interaction structure rather than different access to external evidence.

For the main runs, the scaffolded single-agent setting allows up to four passes per section with critique–retrieval–revision triples. The zero-shot and free-refinement single-agent settings run for up to 19 refinement rounds. The domain-agnostic multi-agent setting uses three generic research agents, six utterances per round, and synthesis after every three utterances for 10 rounds. The domain-specific multi-agent setting uses idea-catalyst challenge metadata, with up to three source-domain agents per challenge and one utterance per domain agent before synthesis.

H Simulating Agentic Ideation

We design five ideation pipelines to study how different interaction structures shape the evolution of research ideas. Each pipeline starts from the same input research problem and produces a sequence of proposal versions $y_1, \ldots, y_T$. After each version, IDEA-EVOLUTION maps the proposal into an idea graph G_t, allowing us to compare not only final proposal quality but also how assumptions, methods, evidence, and limitations change across the trajectory.

Our pipeline design varies three factors: the amount of procedural scaffolding, the locality of each revision, and the source of critique. Procedural scaffolding controls how much the system constrains the writer's next action. Revision locality determines whether each turn modifies a named section, an agent-selected section, or the full proposal. The source of critique distinguishes self-revision from deliberation among multiple agents, including agents with explicitly assigned disciplinary perspectives. Together, these settings create trajectories with different degrees of experimental control and different opportunities for conceptual development.

Single-Agent Scaffolded. In the scaffolded setting, a single writer follows a fixed proposal template and revises one named section at each turn. The section order is predetermined, so the system knows which part of the proposal is expected to change before the graph update is computed. This setting gives the highest control over the trajectory: when a graph node or edge changes after a turn, that change can be cleanly aligned with the targeted section. We use this pipeline to test whether IDEA-EVOLUTION can capture localized conceptual edits rather than treating every proposal revision as a global rewrite.

Single-Agent Zero-Shot Refine. In the zero-shot refinement setting, a single writer first generates a complete proposal from the research problem, then alternates between critique and revision over the full proposal. Unlike the scaffolded setting, the model is not constrained to edit a particular section. This produces a more natural self-refinement trajectory while preserving a simple single-agent structure. We use this pipeline to study whether unconstrained self-critique leads to substantive graph changes, such as strengthening evidence, narrowing claims, or resolving limitations, or whether it mainly produces surface-level elaboration.

Single-Agent Free Refine. In the free refinement setting, a single writer chooses the next edit target at each step. The model may decide to revise the problem framing, method, experiment plan, limitations, or another proposal component. This setting reduces procedural control but reveals the model's own revision priorities. In particular, it lets us test whether self-directed revision naturally returns to weak or underdeveloped components, or whether the trajectory drifts toward easier-to-elaborate parts of the proposal.

Multi-Agent Domain-Agnostic. In the domain-agnostic multi-agent setting, several generic research agents discuss the proposal before a synthesizer produces the next version. The agents are not assigned specialized disciplinary roles; instead, they provide broad critiques, alternatives, and concerns. The synthesizer then incorporates the discussion into a full-proposal revision. This setting tests the effect of open-ended deliberation: whether multiple generic perspectives introduce new assumptions, alternatives, or limitations that a single agent would miss.

Multi-Agent Domain-Specific. In the domain-specific multi-agent setting, the proposal is reviewed by a panel of agents assigned different disciplinary or methodological perspectives. Each agent critiques the proposal through its assigned lens, such as theoretical framing, empirical design,

implementation feasibility, or evaluation. A synthesizer then selects which proposal components to revise and produces the next version. Compared with the domain-agnostic setting, this pipeline provides more structured diversity: critique is still distributed across agents, but the perspectives are intentionally differentiated. We use this setting to examine whether disciplinary specialization leads to more targeted edits and richer graph evolution around assumptions, methods, and evidence.

I Human Study Protocol

I.1 Research Abstract Submission Form

Figure 4 shows the form used to recruit participants and collect their submitted abstracts. Each participant provided informed consent, demographic information, and one or more research abstracts that were used as inputs to the idea graph construction pipeline.

I.2 Evaluation Interface

Figures 5–13 show the evaluation interface in the order participants encountered it: file upload and graph view (Figures 5–6), six subgraph ratings (Figures 7–11), and overall rating and submission (Figures 12–13).

I.3 Participant Demographics & Research Problems

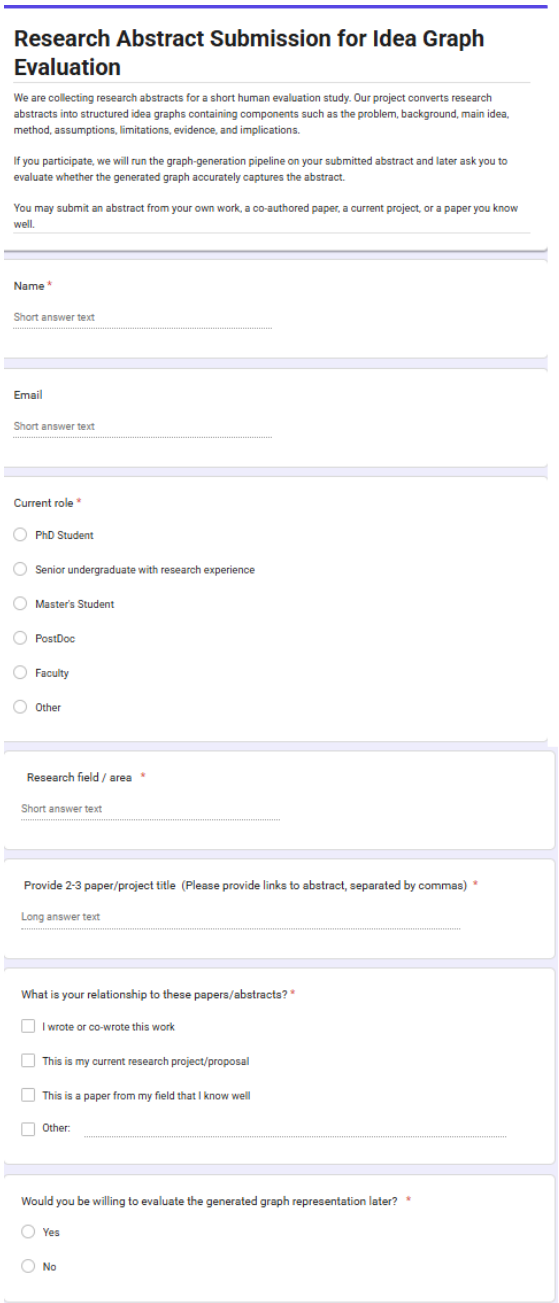

Figure 4: Research abstract submission form.

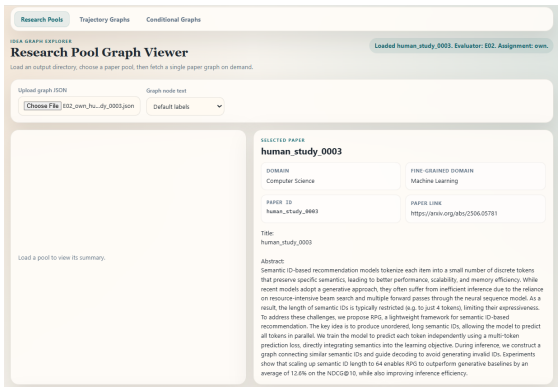

Figure 5: File upload landing screen.

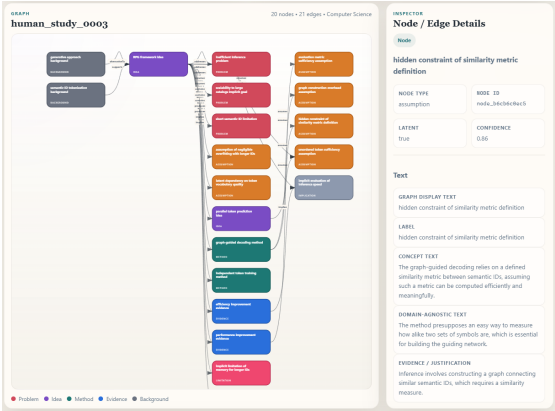

Figure 6: Abstract and constructed idea graph.

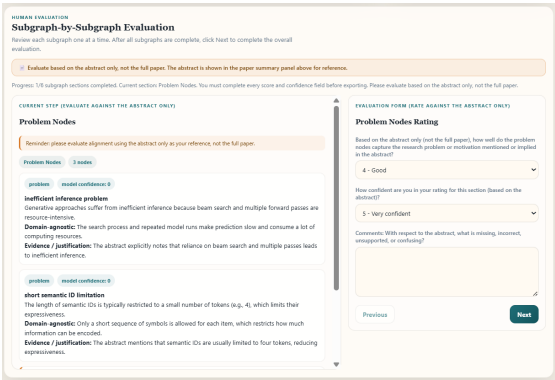

Figure 7: Problem nodes rating.

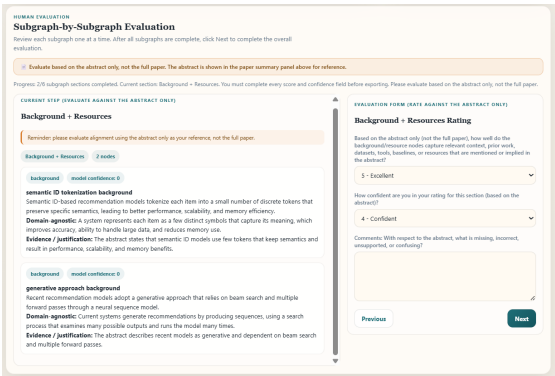

Figure 8: Background and resources rating.

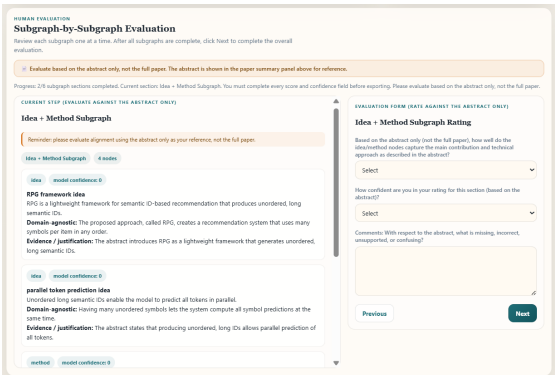

Figure 9: Idea and method rating.

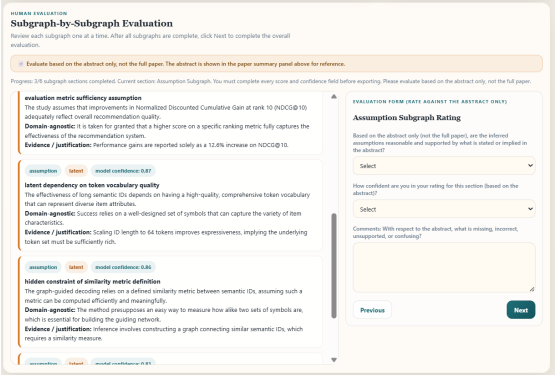

Figure 10: Assumption rating.

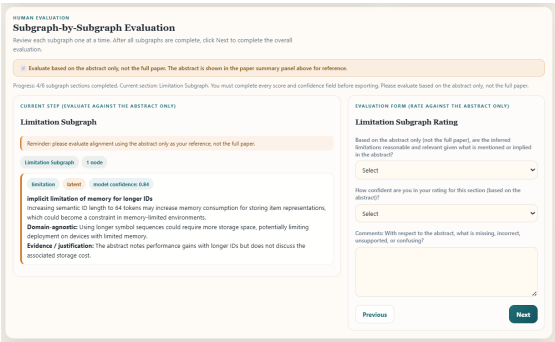

Figure 11: Limitation rating.

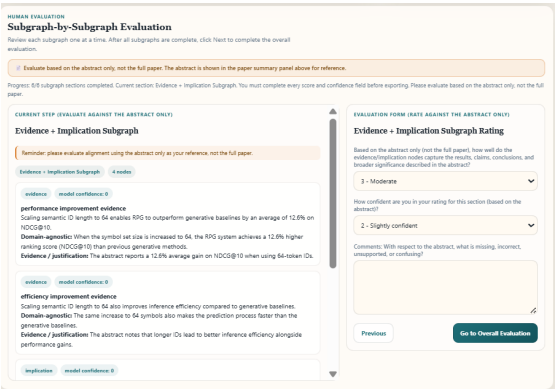

Figure 12: Overall rating and consent.

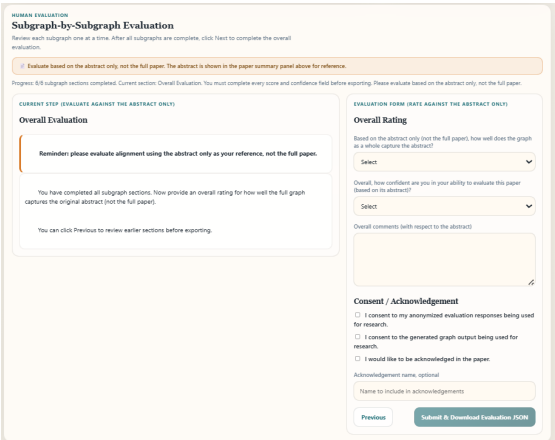

Figure 13: Submission and export.

Evaluator	Background	Primary Research Area	Research Problem Description
P01	PostDoc	Bioinformatics	How can LLM-based agents reduce hallucinations in gene-set analysis, and how can temporal network embeddings capture long-range structural dynamics?
P02	PhD	Machine Learning	How can semantic ID representations and LLM encoders be improved for scalable and transferable recommendation systems?
P03	PhD	Human-Computer Interaction	How can LLMs and prompt optimization improve root cause analysis and incident management in large-scale cloud systems?
P04	PhD	Human-Computer Interaction	How do financial events shape emotional expression in online communities, and how can automated agent modeling enable scalable theory of mind reasoning?
P05	PostDoc	Neuroscience	How can biodegradable nanoparticles be engineered for targeted gene delivery and immune reprogramming in brain tumors?
P06	PostDoc	Computer Vision	How well can multimodal LLMs interpret medical imaging tasks, and what architectures best support 3D medical image understanding?
P07	PhD	Machine Learning	How can privileged information and sentence-level interpretability methods improve machine learning model performance and reasoning transparency?
P08	PhD	Computer Vision	How can reinforcement learning from human feedback be scaled stably for visual content generation across diffusion and flow-based models?
P09	PhD	Natural Language Processing	How can LLM-based systems improve evidence-grounded medical research, personalized health communication, and clinical trial recruitment?
P10	PhD	Natural Language Processing	How can consciousness be formally defined in physically operational terms, and how can LLMs support weakly-supervised hierarchical text classification and concept-based scientific paper retrieval?
P11	PhD	Natural Language Processing	How can LLM-based multi-agent debate, retrieval-augmented claim analysis, and interdisciplinary inspiration frameworks support scientific reasoning and discovery?
P12	Medical Student	Medicine	How do biomarkers and imaging findings complement each other in diagnosing acute heart failure and intermediate-risk pulmonary embolism?
P13	PhD	Bioinformatics	How can microbiome-driven subtyping reveal novel clinical phenotypes and predict disease severity in inflammatory bowel disease?
P14	PhD	Human-Computer Interaction	How can LLMs support hidden-concern reasoning in medical dialogue and enable effective knowledge distillation across model scales?
P15	PhD	Knowledge Distillation	How can knowledge distillation transfer tree-of-thought reasoning capabilities from large to small language models efficiently?

Table 12: **Participant backgrounds for the human study.** Each participant is identified by an anonymized ID. Roles include PhD students, postdoctoral researchers, and a medical student, spanning domains including computer science, bioinformatics, neuroscience, and medicine.