FROM BLACK-BOX JUDGMENTS TO AUDITABLE GRAPHS: STRUCTURED IDEA GENERATION AND EVALUATION VIA SCIENTIFIC KNOWLEDGE GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in large language models (LLMs) have fueled growing interest in autonomous research-idea generation. Now that many systems can propose ideas, candidates are no longer hard to come by; what is hard is deciding which of them justify the cost of implementation. This shifts attention to idea evaluators and raises a more fundamental question: how should these evaluators themselves be evaluated? Existing benchmarks rely on two unreliable signals: black-box LLM judgments that are unauditable and prompt-sensitive, and proxy labels, such as peer-review scores, publication time, and expert annotations, that are validated against human agreement rather than against actual implementation outcomes. We present SciGraph, a graph-based pipeline for generating and evaluating research ideas. SciGraph decomposes historical literature into meta, mechanism, and protocol nodes and connects them through citation, mechanism-combination, and protocol-result comparison relations. Candidate ideas are randomly sampled from a predefined space and rendered as text. Two fixed policies—mainstream and novelty-oriented—then select candidates for execution, differing only in how they weight three graph-derived priors. To evaluate an idea, SciGraph maps its text back to graph nodes and scores it using the same priors. The pipeline also yields SciGraph-Probe, a probe set for assessing idea evaluators. It contains auditable graph representations of 184 papers across 10 agent-related domains, comprising 184 meta nodes, 490 mechanism nodes, and 727 protocol records connected by more than 1,700 edges. It also includes 100 randomly sampled candidate ideas that undergo code synthesis and execution, producing effectiveness labels withheld from all evaluators, together with external bibliometric reference orders for novelty and importance. We conduct two separate experiments. On three idea-generation benchmarks, using the context provided by each benchmark, the two policies show the expected differences. On SciGraph-Probe, SciGraph ranks first on all three dimensions—effectiveness, novelty, and importance—and every score it produces is traceable to individual nodes and edges.

1 INTRODUCTION

Autonomous research systems have recently made rapid progress in generating hypotheses, conducting experiments, and writing papers, gradually closing the loop from literature review to experimental execution. Representative systems include The AI Scientist (Lu et al., 2024), CycleResearcher (Weng et al., 2025), The AI Scientist-v2 (Yamada et al., 2025), Co-Scientist (Gottweis et al., 2026), and, more recently, DeepScientist (Weng et al., 2026). These systems show that LLMs can already support literature reading, research-idea generation, experiment execution, paper writing, and automated reviewing.

As these systems become capable of executing experiments, a growing body of work has focused specifically on research-idea generation, including ResearchAgent (Baek et al., 2025), Chain of Ideas (Li et al., 2025), SciAgents (Ghafarollahi & Buehler, 2024), SciPIP (Wang et al., 2025), Many Heads / VIRSCI (Su et al., 2025), Graphs of Research (GoR) (Gao et al., 2026), Deep Ideation (Zhao et al., 2025), and Evolving Idea Graphs (EIG) (Dong et al., 2026). These methods improve ideation through literature graphs, citation-evolution chains, multi-agent collaboration, and concept networks.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Candidate ideas are therefore no longer hard to come by; what is hard is deciding which of them justify the cost of implementation. Yet no generally accepted tool exists for making this judgment. This places the burden on idea evaluators and raises a more fundamental but less studied question: how should these evaluators themselves be evaluated?

Current idea evaluation relies mainly on two unreliable signals. The first is black-box LLM judgment: five widely used idea-evaluation benchmarks (Qiu et al., 2025; Guo et al., 2024; Shen et al., 2025; Ruan et al., 2026; Mathur et al., 2026) score ideas using LLM prompts. These scores are unauditable and sensitive to prompt wording and candidate order (Si et al., 2025; Ho et al., 2026; Gupta & Pruthi, 2025). The second is proxy labeling: recent evaluation systems use peer-review scores, publication time, or expert annotations as substitutes for idea quality. For example, CycleReviewer (Weng et al., 2025) is trained on review data to provide feedback or serve as a reward model, while DeepScientist (Weng et al., 2026) scores generated papers with DeepReviewer-14B. Validated through agreement with human reviewers or annotators, such systems can be shown to approximate peer review. **What they do not report is how their scores relate to the outcomes of implementing an idea**, and we are not aware of an idea-evaluation resource that pairs candidate ideas with execution outcomes: the signal needed to audit an evaluator is missing.

This gap is structural. Idea evaluation currently operates on unstructured text: a judge reads a paragraph and returns a score, leaving no explicit evidence to inspect when that score is wrong. What is missing is a representation that makes an idea an explicit object—the mechanisms it uses, the papers that carry them, and the protocol used to test it. Rather than proposing another scoring prompt, we therefore structure the object being evaluated. We call a score *auditable* if a reader can identify the nodes, edges, and paper rows that produced it and recompute it from that evidence.

We build on methods that use knowledge graphs and literature graphs for academic-graph retrieval, citation-chain organization, or novelty assessment (Vasu et al., 2025; Buehler, 2025; da Silva et al., 2025). Unlike approaches that model citation relations, entity co-occurrence, or mechanism concepts in isolation, SciGraph unifies paper metadata, mechanisms, and experimental protocols in a single representation.

SciGraph decomposes reference papers into unit subgraphs containing meta, mechanism, and protocol nodes. It then links these subgraphs through citation, combination, and performance edges to form a scientific knowledge graph that captures citation relations, mechanism combinations, and protocol-result comparisons. Candidate ideas are randomly sampled into a shared pool from a predefined mechanism–task–protocol space and rendered as text. Two fixed policies—mainstream and novelty-oriented—then select candidates for execution, differing only in how they weight three graph-derived priors. The same priors serve as the evaluator. Every evaluator receives the same input—the candidate text with its source papers—and SciGraph differs only in converting that literature into the graph internally, resolving the idea to graph nodes, and reading Val for effectiveness classification and Nov and Imp for novelty and importance ranking. Given this mapping its three outputs are deterministic functions of the graph, so the comparison isolates how the evidence is represented rather than how much of it each method receives.

To test whether these graph-derived scores are informative, we assemble SciGraph-Probe, a probe set produced as a by-product of the same pipeline. It contains 184 provenance-linked papers covering 10 agent- and LLM-related domains and 100 candidate ideas selected by the two policies from randomly sampled per-domain pools. Each idea is implemented as runnable code and executed against a task-specific baseline, grounding effectiveness in observed outcomes rather than a judge's score. Of these candidates, 46 improve on their baseline and 54 do not, so the decision an evaluator faces is non-trivial in both directions. For novelty and importance, which lack directly measurable ground truth for unpublished ideas, we use established bibliometric indicators as external reference orders. These indicators are withheld from all evaluators and described in Appendix B. Our comparison with prior idea-generation methods is a separate experiment on different candidates: every system consumes the materials its benchmark provides, SciGraph builds its graph from those same materials, and none of those ideas is executed.

Our contributions are as follows:

1) We assemble SciGraph-Probe, a probe set whose effectiveness labels come from execution rather than from a score: 100 candidates selected by the two policies from randomly sampled per-domain pools are synthesized into runnable code and run against task-specific baselines, over 184 provenance-

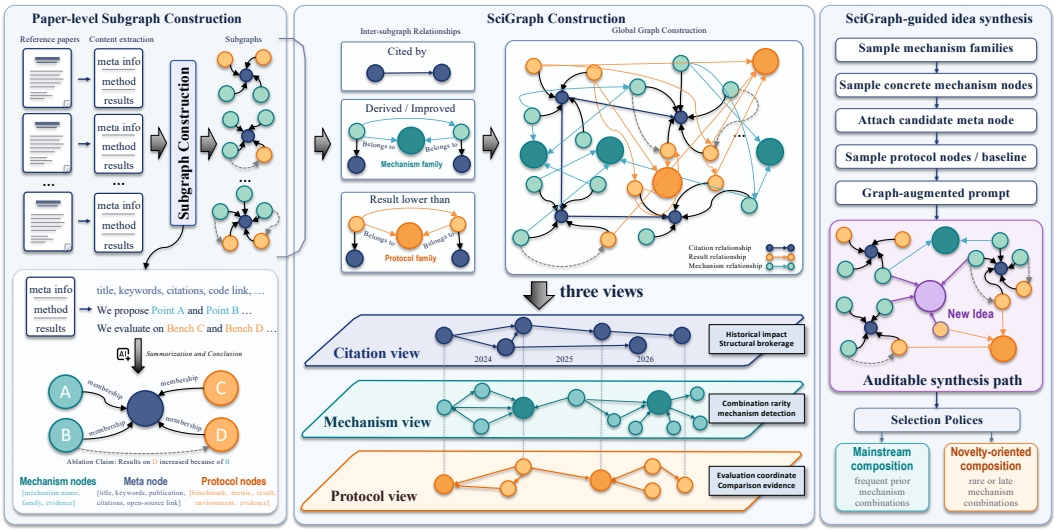

Figure 1: Overview of SciGraph. Reference papers are decomposed into typed subgraphs and linked to form a heterogeneous graph with three views. Candidate ideas are randomly sampled into a shared pool and rendered as text, and two fixed policies select which candidates to execute. For evaluation, an idea's text is mapped back to the graph to obtain auditable prior-based scores.

linked papers in 10 domains. Novelty and importance use established bibliometric indicators as external reference orders, calibrated against ex-post citation percentiles on the same corpus and withheld from every evaluator;

2) Using this probe set, we show that current evaluation practice is weak where it is most often applied: four LLM judges reach 0.520–0.600 accuracy against a 0.54 majority-class rate while labelling almost every candidate effective, and their novelty rankings change sign across backbones. Across benchmark-native evaluators the novelty rank correlation spans 0.27 in τ, more than it varies across the ideas themselves;

3) We show that a scientific knowledge graph can serve as the scoring substrate for an unpublished idea. SciGraph jointly models paper metadata, mechanisms, and experimental protocols, so a candidate's target protocol carries historical baseline comparisons and not citations alone. From those same inputs, SciGraph resolves an idea to graph nodes, and given that resolution the three ex-ante scores and the selection decision are deterministic functions of named nodes, edges, and paper rows.

2 Related Work

2.1 Structured Approaches to Research Idea Generation

Recent autonomous research systems increasingly automate the research loop from idea generation to experiment execution, paper writing, and review (Lu et al., 2024; Weng et al., 2025; Yamada et al., 2025; Gottweis et al., 2026). Review scores guide these systems, but whether such a score predicts an idea's effectiveness once implemented is not separately validated.

A parallel line of work grounds ideation in structured literature, using it to widen the context available to a generator, to impose temporal order, or to supply training signal. ResearchAgent expands a citation neighbourhood and entity store around a core paper (Baek et al., 2025), and SciPIP likewise retrieves literature-derived elements before proposing (Wang et al., 2025); Chain of Ideas builds time-evolving chains and compares ideas pairwise (Li et al., 2025); SciAgents samples long-range paths from an ontology graph (Ghafarollahi & Buehler, 2024); GoR uses citation-evolution graphs as supervision (Gao et al., 2026); Deep Ideation adds concept networks with trained critics (Zhao et al., 2025); EIG builds claim-support graphs of the proposal itself (Dong et al., 2026); HypER uses provenance-aware paths to separate logical dependence from surface similarity (Vasu et al., 2025);

and Graph-PReFLexOR incorporates graph construction into model reasoning (Buehler, 2025). In each case the graph is an input or training representation for *producing* a proposal; how good the proposal is gets settled afterwards by a separate judgment.

2.2 Reference Signals for Research Idea Evaluation

LLM judges are widely used in idea benchmarks. AI Idea Bench 2025, IdeaBench-research, IDR-Bench, LiveIdeaBench, and SciAidanBench all score ideas with LLM prompts, along dimensions such as alignment, overlap, novelty, feasibility, and coherence (Qiu et al., 2025; Guo et al., 2024; Shen et al., 2025; Ruan et al., 2026; Mathur et al., 2026). Such comparisons are useful, but LLM-based scores remain vulnerable to position effects, self-evaluation bias, and unreliable novelty judgments (Si et al., 2025; Ho et al., 2026; Gupta & Pruthi, 2025). Human studies are a stronger reference but expensive, and still do not establish whether a proposed idea works when implemented (Si et al., 2025).

Evaluation can also draw on external reference signals: peer-review scores (CycleReviewer (Weng et al., 2025) and SciNova, which accompanies GraphMind (da Silva et al., 2025)); publication time (SchNovel (Lin et al., 2025)); expert annotations (soundness benchmarks (Ho et al., 2026)); and the rediscovery of hypotheses already published, as in MOOSE-Chem for chemistry (Yang et al., 2025). Each is validated against its own supervision, showing only that it approximates the decision it imitates; none reports how it relates to what happens when an idea is implemented. We are not aware of a resource that pairs candidate ideas with execution outcomes.

Two systems are particularly relevant because they also aim at traceable judgments. GraphMind assesses novelty from a paper's internal structure and a graph of cited and semantically related papers (da Silva et al., 2025), and OpenNovelty produces evidence-linked novelty reports at scale, deployed on over 500 conference submissions (Zhang et al., 2026). Both score work that is already written, address novelty alone, and are checked against expert judgment; we score candidate ideas before they are implemented, across effectiveness, novelty, and importance, and check them against execution outcomes.

SciGraph separates scoring from validation. Its graph is neither a retrieval context nor a learned predictor but a scoring substrate whose scores decompose into inspectable nodes and edges. The reference orders are computed outside the pipeline, from execution outcomes and bibliometric indicators, and withheld from every evaluator including ours.

3 Method

3.1 SciGraph construction pipeline

SciGraph takes a set of reference papers as input and constructs a graph used for candidate evaluation (Figure 1). These are the same reference papers that every method we compare against receives; graph construction is an internal step of SciGraph rather than an additional input. Candidate sampling is a separate operation: candidates are drawn from a predefined mechanism–task–protocol space without using the prior scores. What distinguishes SciGraph is therefore not the literature it sees but the explicit structure it derives from it. Proposing a candidate requires specifying mechanisms and a target protocol, while evaluating it requires evidence about novelty, importance, and historical protocol performance. SciGraph accordingly decomposes each paper into three basic elements, extracted from different parts of the paper, each retaining its textual evidence. For each paper p, SciGraph constructs a unit subgraph

$$S(p) = \big(v_p^{\mathrm{meta}}, V_p^{\mathrm{mech}}, V_p^{\mathrm{prot}}, E_p\big).$$

Typed nodes. The *meta* node records properties of the paper itself—title, keywords, citation information, and available open-source or implementation links—and acts as the paper-level anchor for the other nodes in $S(p)$. A *mechanism instance* is paper-specific: it stores the mechanism name used in that paper, a short description, and the supporting text span. We retain the principal two or three mechanisms used by a paper rather than treating every implementation detail as a separate mechanism. Each instance also has a canonical concept identifier. A *mechanism family* is the canonical concept that groups its paper-specific instances. For example, the graph contains paper-specific instances

of "verifier-guided selection" from Snell et al. (2024) and Lightman et al. (2024); both are linked to the canonical concept `mechanism-concept:verifier-guided-selection`. The family aggregates recurring uses of the concept, while the instances retain the paper-specific wording and evidence. *Protocol* nodes are extracted from the experimental section; each records the protocol or benchmark name, the reported metric such as precision, efficiency, or cost, the numerical result, the direction in which the metric improves, and the paper's own explanation of the result.

Paper-level subgraph. Within $S(p)$, the edge set E_p consists of intra-subgraph and inter-subgraph edges. Intra-subgraph edges connect the meta node to each mechanism and protocol node by membership; in addition, if the paper explicitly states that a mechanism causes or explains an improvement on a protocol result, SciGraph adds a directed mechanism-to-protocol causal edge and stores the supporting text, so that such edges are never inferred from the mere co-occurrence of a mechanism and a result. Inter-subgraph edges link one paper subgraph to another along three relations. Meta nodes are connected by citation: for a citation from p_i to p_j we add a directed edge from $v_{p_j}^{\mathrm{meta}}$ to $v_{p_i}^{\mathrm{meta}}$, and temporal-consistency checks reject time-reversed citations so that the citation relation forms a time-oriented directed acyclic graph. Mechanism nodes are connected by combination and refinement: mechanisms that co-occur in a paper induce a combination relation with the source paper retained as evidence, and mechanism nodes with the same or similar meaning are grouped under a family node that aggregates a line of mechanisms across papers. Protocol nodes are connected by result comparison: comparable protocol nodes are ordered by their reported results, with the lower-performing result pointing to the higher-performing one after accounting for metric direction, and protocol nodes measuring the same benchmark protocol are grouped under a protocol family node.

Three views. Let $G = (V, E)$ denote the resulting graph. Its three views are the induced subgraphs

$$G_{\mathrm{cite}} = G[V^{\mathrm{meta}}], \qquad G_{\mathrm{mech}} = G[V^{\mathrm{mech}} \cup V^{\mathrm{fam}}_{\mathrm{mech}}], \qquad G_{\mathrm{prot}} = G[V^{\mathrm{prot}} \cup V^{\mathrm{fam}}_{\mathrm{prot}}], \tag{1}$$

and each contributes evidence to candidate evaluation. The citation view records the temporal development of a field and supplies the influence of the papers on which a candidate is built. The mechanism view exposes individual mechanisms, their families, and their recurring combinations, allowing the evaluator to measure how unusual a sampled combination is relative to the literature. The protocol view records comparable experimental results within each protocol family and supplies the historical performance evidence for a candidate's target protocol.

3.2 Idea evaluation method

For each domain, we first sample $N = 30$ candidates from the predefined mechanism–task–protocol space. The mainstream and novelty-oriented policies rank this shared pool using the graph priors below and each select five candidates. If a candidate is selected by both policies, it is assigned to the policy under which it has the higher rank, and the other policy selects its next highest-ranked unassigned candidate. Each candidate is mapped after sampling to a mechanism set $M(c)$ and target protocol g.

Candidate-to-graph mapping and evaluation. For each idea in the 100-candidate probe, SciGraph takes the candidate text and its source papers—the same input the other evaluators receive. An LLM first decomposes the proposed method into mechanism mentions. Each mention is matched to a canonical mechanism family, which retrieves the family instances, evidence spans, carrier papers, and protocol records. The target protocol is matched by benchmark name. A mechanism that cannot be matched to a family is retained as out-of-graph rather than silently replaced by a nearby mechanism. SciGraph then uses the resolved graph evidence to compute the same three priors used in candidate selection: Val for effectiveness classification, and Nov and Imp for novelty and importance ranking. The other evaluators classify or rank the same material directly, without this resolution step, so the two differ in how the evidence is represented rather than in what is available to them. Given the resolved mapping, SciGraph's three outputs are deterministic graph calculations. SciGraph then reads three ex-ante priors from the constructed graph:

$$\mathrm{Nov}(c) = 0.8R_{\mathrm{pair}}(c) + 0.2R_{\mathrm{out}}(c), \qquad \mathrm{Imp}(c) = 0.4I_{\mathrm{bridge}}(c) + 0.6I_{\mathrm{ground}}(c),$$

$$\mathrm{Val}(c, g) = \frac{\sum_{p \in H(c,g)} \mathrm{sim}(c, p) y(p, g) + 3\bar{y}(g)}{\sum_{p \in H(c,g)} \mathrm{sim}(c, p) + 3}.$$

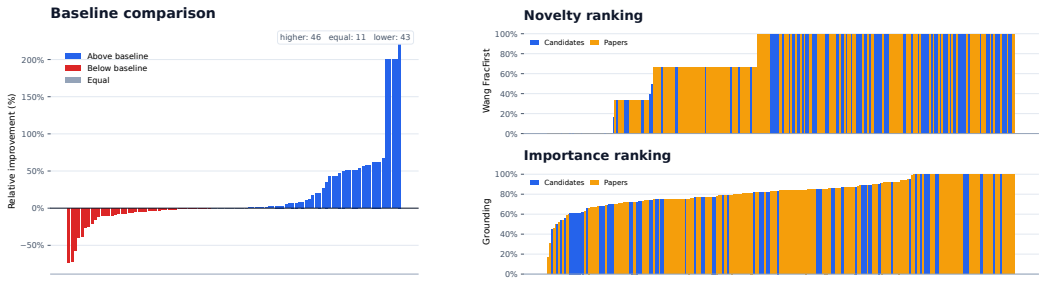

Figure 2: The 100 policy-selected and executed candidates against their baselines (left) and ranked jointly with the 184 published papers under the novelty and importance indicators (right); blue denotes candidates and orange denotes real papers.

Nov is associated with the rarity of historical mechanism combinations; Imp with mechanism bridging and the citation standing of carrier papers; and Val with whether historically similar methods beat their baselines on the target protocol. For Table 2, SciGraph predicts effective when $\mathrm{Val}(c, g) \geq 0.5$ and ineffective otherwise; this fixed threshold is not tuned on the execution labels. These priors are distinct from the post-hoc reference orders used to evaluate evaluators. Mainstream and novelty-oriented policies apply task-normalized weights $(0.2, 0.4, 0.4)$ and $(0.6, 0.2, 0.2)$, respectively, to rank the shared candidate pool and select which candidates are executed. Definitions, coverage rules, and missing-value handling are in Appendix B.3.

3.3 SciGraph-Probe construction and evaluation

To examine whether graph-derived scores are informative, we assemble SciGraph-Probe, a probe set that couples idea evaluation to real execution outcomes and to external literature indicators. We collected 184 papers published in recent years across 10 popular agent- and LLM-related domains, drawn from top venues and arXiv. For each domain, we build a SciGraph, sample a shared pool of $N = 30$ candidates, and select five candidates under each policy, yielding 100 executed candidate ideas in total (10 domains $\times$ 10 ideas). For every candidate, we synthesize runnable code with CodeBuddy and execute real experiments under a fixed task protocol with a task-specific baseline. The resulting graph contains 184 meta nodes, 490 mechanism nodes, and 727 protocol records, connected by more than 1,700 edges.

Reference orders. The three reference orders are computed *outside* the pipeline and are invisible to every evaluator, including SciGraph itself; they exist only to score evaluators. Effectiveness comes from the real execution outcome relative to the task baseline—a candidate whose executed result exceeds its baseline is effective—which yields 46 effective and 54 ineffective candidates. Novelty and importance have no physical ground truth for an unpublished idea, so we borrow two established bibliometric indicators instead of inventing our own. For novelty we use the first-combination fraction FracFirst of Wang et al. (2017), the fraction of mechanism pairs never co-occurring before the proposal cutoff. FracFirst is computed directly from the original-paper corpus and is distinct from the constructed graph's weighted combination-edge prior Nov; it is never an input to SciGraph. For importance we use citation Grounding, the citation-percentile strength of the historical papers carrying a candidate's mechanisms, following the bibliometric grounding view of Bornmann et al. (2010) with the aggregation of Mammola et al. (2021). Both indicators are computed under strict pre-cutoff filtering; their definitions, missing-value rules, and calibration against subsequent citation percentiles on the 184 real papers are given in Appendix B. Exploratory alternatives—Uzzi-style z-scores (Uzzi et al., 2013), combination age (Mukherjee et al., 2017), protocol Anchor, and mechanism Between—are reported there but are not used as reference orders.

At evaluation time every evaluator receives the same input: the candidate text and its source papers. LLM and benchmark-native evaluators score that input directly, whereas SciGraph first resolves the candidate to the graph and then computes Val for effectiveness classification and Nov and Imp for novelty and importance ranking. No evaluator receives the execution results or the reference orders. Judges predict effective or ineffective, scored by accuracy, precision, recall, and F1; and rank the nominal five candidates within each of 20 task–benchmark groups, scored by group-averaged Kendall τ, Spearman ρ, NDCG@5, and Top-1 hit rate.

Table 1: Generation results on three existing idea-generation benchmarks. All systems consume the context each benchmark provides; SciGraph builds its graph from that same context and produces the number of ideas the benchmark requires. These candidates are separate from the 100 executed candidates of Section 3.3. Benchmark names are shown in the first header row, followed by task metrics. Higher is better; best and second-best values are shown in bold and underlined, respectively. The two policy rows use different fixed selection criteria after candidate generation.

	IdeaBench				AI Idea Bench 2025						IDRBench
Method	Sem. Sim.↑	Overlap↑	Nov. Insight↑	Fea. Insight↑	I2T Mot.↑	I2T Exp.↑	I2I Mot.↑	I2I Exp.↑	IMCQ Mot.↑	IMCQ Exp.↑	Comp.↑ (%)
Native Prompt	0.529	4.687	**6.823**	5.863	**3.923**	3.742	3.692	3.651	**0.293**	0.226	74.1
zero-shot	0.455	4.037	5.407	5.042	3.427	3.371	3.315	3.274	0.107	0.137	65.7
few-shot	0.476	4.224	5.603	5.276	3.586	3.543	3.472	3.432	0.128	0.168	69.1
CoT	0.503	4.456	5.802	5.574	3.703	3.644	3.573	3.536	0.147	0.192	74.4
RAG	<u>0.535</u>	6.904	5.906	5.728	3.884	3.815	3.754	3.716	0.235	0.243	75.8
GoR	0.534	6.857	6.214	6.464	3.872	3.868	3.837	3.755	0.223	0.263	78.1
GraphMind	0.528	<u>7.146</u>	6.251	6.357	<u>3.898</u>	3.895	<u>3.884</u>	3.772	0.213	0.275	77.2
Deep Ideation	0.534	6.536	6.583	6.286	3.871	3.833	3.775	3.733	0.213	0.252	76.9
GoAI	0.524	6.855	6.446	6.031	3.821	3.776	3.714	3.674	0.189	0.234	75.9
Dolphin	0.516	5.278	3.847	6.423	3.807	3.916	3.796	3.797	0.178	0.306	75.5
AI-Scientist	0.518	5.343	3.913	6.472	3.845	3.978	3.815	3.838	0.248	0.313	77.1
DeepScientist	0.522	5.358	4.028	<u>6.483</u>	3.852	3.947	3.862	3.815	0.205	<u>0.324</u>	77.8
Ours (Mainstream policy)	**0.542**	**7.454**	4.062	**7.814**	3.857	<u>4.006</u>	**3.906**	<u>3.866</u>	0.231	0.294	**82.5**
Ours (Novelty policy)	0.532	6.535	<u>6.643</u>	6.184	3.814	**4.024**	3.733	**3.884**	<u>0.273</u>	**0.332**	<u>79.4</u>

Figure 2 places the 100 candidates next to the 184 published papers. The left panel reports each candidate's executed outcome relative to its baseline: 46 exceed their baselines, 11 are tied, and 43 fall below, so roughly half of the selected candidates produce a measurable gain under a real protocol. The two right panels rank candidates and published papers together under the same indicator, with missing values omitted rather than imputed. The two populations interleave along the whole range rather than separating into a candidate band and a paper band, on both novelty and importance. This overlap is evidence only under these two indicators; it does not establish that the selected candidates are generally equivalent to published work. Together with the 46 executed gains, it shows that useful candidates are present in the sampled pool. The remaining problem is to identify them before execution: indicator position and executed gain are different things, which is why the three reference orders are kept separate.

4 Experiments

4.1 Setup

We evaluate generation on IdeaBench-research (Guo et al., 2024), AI Idea Bench 2025 (Qiu et al., 2025), and the I3 paper-pair task of IDRBench (Shen et al., 2025), using their native metrics.

For generation, we compare direct prompting (zero-shot, few-shot, and CoT), retrieval augmentation (RAG), graph-based methods (GoR (Gao et al., 2026), GraphMind (da Silva et al., 2025), Deep Ideation (Zhao et al., 2025), and GoAI (Gao et al., 2025)), autonomous research systems (Dolphin (Yuan et al., 2025), AI-Scientist (Lu et al., 2024), and DeepScientist (Weng et al., 2026)), and each benchmark's native prompt. All systems consume the materials their benchmark provides; SciGraph converts those same materials into a graph as an internal step, samples and renders candidates, and applies the mainstream or novelty-oriented policy to select the number of ideas the benchmark requires. The policies are selection rules applied after candidate generation, not alternative generators. All LLM-dependent methods use dsv4-flash (dsv4f) as the generation backbone. We report each benchmark's native generation metrics: Semantic Similarity, Idea Overlap, and Novelty/Feasibility Insight on IdeaBench-research; I2T, I2I, and IMCQ motivation/experience on AI Idea Bench 2025; and I3 composition on IDRBench.

Comparison scope. Table 1 compares backbone-controlled systems that use the same task-provided context; interface-mismatched systems are excluded. These benchmark-derived candidates are separate from the 100 executed SciGraph-Probe candidates and are not executed. For evaluation, we compare SciGraph, LLM judges, and benchmark-native evaluators on SciGraph-Probe under the protocol and metrics of Section 3.3; we additionally apply the LiveIdeaBench evaluator, although its keyword-input task is unsuitable for generation.

Table 2: Unified evaluation results for effectiveness, novelty, and importance. The header row gives the evaluation dimensions and the following row gives the metrics. Higher is better; "—" denotes an unavailable result.

	Effectiveness				Novelty				Importance			
Evaluator / Method	Acc.↑	Prec.↑	Rec.↑	F1↑	Kendall τ	Spearman ρ	NDCG@5	Top-1	Kendall τ	Spearman ρ	NDCG@5	Top-1
dsv4-flash (LLM Judge)	0.600	0.538	0.935	0.683	+0.120	+0.160	0.979	0.15	+0.100	+0.125	0.915	0.20
kimi-k2.6 (LLM Judge)	0.560	0.511	0.978	0.672	+0.170	+0.195	0.981	0.30	+0.080	+0.120	0.905	0.15
glm-5.2 (LLM Judge)	0.520	0.489	0.978	0.652	+0.200	+0.215	0.980	0.35	−0.010	+0.010	0.898	0.00
minimax-m2.7 (LLM Judge)	0.520	0.489	1.000	0.657	−0.090	−0.150	0.968	0.15	−0.010	0.000	0.910	0.15
LiveIdeaBench (Bench Evaluator)	—	—	—	—	+0.010	+0.015	0.900	0.20	—	—	—	—
IdeaBench-research (Bench Evaluator)	—	—	—	—	+0.240	+0.275	0.955	0.30	—	—	—	—
AI Idea Bench (Bench Evaluator)	—	—	—	—	−0.030	−0.035	0.890	0.10	+0.050	+0.060	0.905	0.15
Ours (SciGraph)	**0.700**	**0.611**	**0.957**	**0.746**	**+0.280**	**+0.340**	**0.985**	**0.40**	**+0.300**	**+0.350**	**0.940**	**0.30**

Supplementary analyses. Appendix A provides the detailed comparison scope (Appendix A.1), per-family generation analysis, and generation-side ablations; Appendix B provides the reference-order definitions and calibration results, together with traceable execution case studies (Appendices B.6 and B.6).

4.2 Experiment 1: generation on existing benchmarks

Table 1 contrasts selection rules on each benchmark-derived pool. Mainstream improves grounding-type metrics (0.542 Semantic Similarity, 7.814 Feasibility Insight, and 82.5 composition) but has lower apparent novelty (4.062). Novelty-oriented reaches 6.643 Novelty Insight at lower Feasibility Insight (6.184); Native Prompt reaches the table maximum of 6.823. Thus graph conditioning reflects a grounding–novelty trade-off, not uniform generation superiority. It does not establish superiority to random selection because excluded candidates are not executed.

4.3 Experiment 2: idea evaluation

Following the protocol of Section 3.3, judges predict effective/ineffective for the 100 executed ideas (46 effective, 54 ineffective) and rank the nominal five candidates within each of 20 task–benchmark groups against the novelty and importance indicators. Table 2 reports all three dimensions together.

LLM judges approach the majority-class baseline and disagree across backbones. The four judges reach 0.520–0.600 accuracy against the 0.54 majority-class rate; recall of 0.935–1.000 at precision 0.489–0.538 shows that they call almost every candidate effective. Novelty τ spans -0.090 to $+0.200$ across models, and benchmark-native evaluators similarly span -0.030 to $+0.240$. NDCG@5 is uniformly high (0.968–0.981) because five-item groups make it weakly discriminative. Their references do report how related methods fared on these benchmarks, but a judge reading them as prose does not aggregate those outcomes into a decision.

Graph-derived scoring ranks first on all three dimensions. SciGraph attains 0.700 accuracy and 0.746 F1 on effectiveness, $\tau = +0.280$ on novelty, and $\tau = +0.300$ on importance. Each score is read from a different part of the graph rather than from one impression of the text: the protocol view makes countable how similar mechanisms historically fared on the target benchmark, the mechanism view counts prior co-occurrences instead of guessing rarity, and the citation view exposes the standing of the papers a combination is built on (Appendix A). The absolute levels remain modest—$\tau \approx 0.3$ is a weak ordering and 0.700 accuracy is far from solved—so the contribution is not a strong predictor but a scoring procedure that decomposes into named nodes, edges, and paper rows. Mechanism resolution is LLM-assisted; given the resolved mapping, the scores use no language model and are deterministic graph calculations.

4.4 Ablations

View and structure signals. Table 3 ablates graph views and graph structure, changing one factor per variant. Removing the mechanism, citation, or protocol view primarily reduces novelty (τ: $+0.280 \rightarrow +0.096$), importance ($+0.300 \rightarrow +0.148$), or effectiveness (0.700 to 0.580), respectively.

Table 3: Graph ablations under one metric block. The first column distinguishes the full evaluation settings, view-removal variants, and structure-control variants; the full SciGraph evaluation is identical to Ours in Table 2. The unstructured judge receives the same candidate text and source papers but no graph. Higher is better.

		Effectiveness		Novelty		Importance	
Group	Variant	Acc.↑	F1↑	τ↑	ρ↑	τ↑	ρ↑
Evaluation settings	Full SciGraph evaluation	0.700	0.746	+0.280	+0.340	+0.300	+0.350
	Unstructured LLM judge	0.600	0.683	+0.120	+0.160	+0.100	+0.125
View removal	w/o citation view	0.650	0.720	+0.243	+0.302	+0.148	+0.185
	w/o mechanism view	0.620	0.703	+0.096	+0.131	+0.225	+0.271
	w/o protocol view	0.580	0.644	+0.268	+0.331	+0.186	+0.224
Structure controls	Evidence shuffle	0.630	0.678	+0.168	+0.207	+0.152	+0.188
	Edge shuffle	0.630	0.673	+0.142	+0.181	+0.128	+0.162
	Random-edge control	0.610	0.661	+0.117	+0.154	+0.105	+0.141

Evidence shuffle perturbs evidence used in resolution and scoring, whereas edge shuffle and random-edge control perturb graph connectivity. The latter reduce ranking to $\tau \approx 0.10$–0.14, the level of an unstructured judge working from the same references, showing that the signal depends on specific graph connectivity. Together with the view removals, this pattern indicates that the three views contribute distinct evidence rather than interchangeable context.

5 Discussion and Limitations

Novelty and importance do not have physical ground truth for unpublished ideas. We therefore adopt established bibliometric indicators as external reference orders rather than defining our own: $\mathrm{FracFirst}$ (Wang et al., 2017) for novelty and citation Grounding (Bornmann et al., 2010; Mammola et al., 2021) for importance. Their definitions, leakage controls, and corpus-level calibration on 184 real papers are given in Appendix B (Tables 5 and 6). "Never observed" means absence of evidence within our declared search window, and mechanism-usage extraction is content-based but carries verbatim evidence per edge.

The graph covers 184 papers, biased toward AI tasks; execution is budget-limited; and candidate-idea importance is an ex-ante quantity because unpublished ideas have no citation history. The calibration supports the selected indicators within this corpus, but does not establish that they are universal measures of future scientific impact. A further caveat concerns what auditability does and does not buy: a score that decomposes into named nodes and edges can be checked and contested, but being inspectable is not the same as being correct, and the absolute agreement levels we report ($\tau \approx 0.3$, 0.700 accuracy) make clear that graph-derived scoring is a weak instrument that happens to be a transparent one.

6 Conclusion

SciGraph makes idea generation and evaluation structured: it links literature metadata, mechanisms, and protocols, then uses fixed graph priors to select candidates from a randomly sampled pool. SciGraph-Probe couples 100 selected candidates to code synthesis and fixed-baseline execution, enabling comparison with LLM judges and benchmark-native evaluators.

The two policies are fixed graph-conditional selection rules, not generators. Given the same inputs, LLM judges approach the majority-class baseline on effectiveness and yield backbone-dependent novelty rankings. Graph scoring is modest but ranks first on all three reported dimensions. Mechanism resolution is LLM-assisted; given the resolved mapping, graph scoring is deterministic and traceable to named nodes, edges, and paper rows. Progress on idea evaluation should therefore prioritize representations whose judgments can be inspected and contested.

USE OF AI-ASSISTED TOOLS

In this work, we used generative AI tools for literature processing, mechanism annotation (evidence-verified per edge), drafting assistance, and code generation for the execution pipeline. All AI-generated annotations were checked against the original papers by the authors; all AI-generated code was executed and verified; and all reported experiments were run by the authors with fixed seeds. We take responsibility for the final content of this work, including text, claims, and code produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The graph construction, candidate representation, execution protocol, and reference-order computations are specified in Section 3 and Appendices B and B.3. We state the node and edge semantics, sampling procedure, scoring equations, missing-value rules, evaluation metrics, and fixed weights required to reproduce the reported computations. In particular, co-occurrence counts, prior scores, the fixed $\mathrm{Val} \geq 0.5$ effectiveness threshold, $\mathrm{FracFirst}$, citation percentiles, and Grounding follow the definitions in Appendix B.3.

ETHICS STATEMENT

This work evaluates research ideas generated and executed by automated systems on public benchmarks; it involves no human subjects and no personally identifiable data. All reference papers are publicly available publications, and all executions use public code and datasets under their original licenses.

REFERENCES

Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. Researchagent: Iterative research idea generation over scientific literature with LLMs. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics*, 2025. URL https://arxiv.org/abs/2404.07738.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024. URL https://arxiv.org/abs/2308.14508.

Lutz Bornmann, Félix de Moya Anegón, and Loet Leydesdorff. Do scientific advancements lean on the shoulders of giants? a bibliometric investigation of the ortega hypothesis. *PLoS ONE*, 5 (10):e13327, 2010. doi: 10.1371/journal.pone.0013327. URL https://doi.org/10.1371/journal.pone.0013327.

Ulrik Brandes. A faster algorithm for betweenness centrality. *The Journal of Mathematical Sociology*, 25(2):163–177, 2001. doi: 10.1080/0022250X.2001.9990249. URL https://doi.org/10.1080/0022250X.2001.9990249.

Markus J. Buehler. In-situ graph reasoning and knowledge expansion using graph-PReFLexOR. *Advanced Intelligent Discovery*, 2025.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heather Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, et al. Evaluating large language models trained on code. In *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Italo Luis da Silva, Hanqi Yan, Lin Gui, and Yulan He. GraphMind: Interactive novelty assessment system for accelerating scientific discovery, 2025. URL https://arxiv.org/abs/2510.15706. EMNLP 2025 Demo paper.

Jiangwen Dong, Bo Li, and Wanyu Lin. Evolving idea graphs with learnable edits-and-commits for multi-agent scientific ideation, 2026. URL https://arxiv.org/abs/2605.04922.

Songyang Gao, Yinghui Xia, Siyi Liu, and Hui Xiong. Graphs of research: Citation evolution graphs as supervision for research idea generation, 2026. URL https://arxiv.org/abs/2605.14790.

Xian Gao, Zongyun Zhang, Ting Liu, and Yuzhuo Fu. GoAI: Enhancing AI students' learning paths and idea generation via graph of AI ideas, 2025. URL https://arxiv.org/abs/2503.08549.

Alireza Ghafarollahi and Markus J. Buehler. Sciagents: Automating scientific discovery through multi-agent intelligent graph reasoning, 2024. URL https://arxiv.org/abs/2409.05556. Also published in Advanced Materials.

Juraj Gottweis, Wei-Hung Weng, et al. Accelerating scientific discovery with Co-Scientist. *Nature*, 2026. doi: 10.1038/s41586-026-10644-y. URL https://doi.org/10.1038/s41586-026-10644-y.

Sikun Guo, Amir Hassan Shariatmadari, Guangzhi Xiong, Albert Huang, Eric Xie, Stefan Bekiranov, and Aidong Zhang. Ideabench: Benchmarking large language models for research idea generation, 2024. URL https://arxiv.org/abs/2411.02429.

Tarun Gupta and Danish Pruthi. All that glitters is not novel: Plagiarism in AI-generated research. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25721–25738, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.1249. URL https://aclanthology.org/2025.acl-long.1249/.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2009.03300.

Sy-Tuyen Ho, Minghui Liu, Huy Nghiem, and Furong Huang. SoundnessBench: Can your AI scientist really tell good research ideas from bad ones?, 2026. URL https://arxiv.org/abs/2605.30329.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.06770.

Long Li, Weiwen Xu, Jiayan Guo, et al. Chain of ideas: Revolutionizing research via novel idea development with LLM agents. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025. URL https://arxiv.org/abs/2410.13185.

Minghao Li et al. Api-bank: A comprehensive benchmark for tool-augmented llms. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://arxiv.org/abs/2304.08244.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, pp. 39578–39601, 2024. URL https://arxiv.org/abs/2305.20050.

Ethan Lin, Zhiyuan Peng, and Yi Fang. Evaluating and enhancing large language models for novelty assessment in scholarly publications. In *Proceedings of the 1st Workshop on AI and Scientific Discovery: Directions and Opportunities*, pp. 46–57, Albuquerque, New Mexico, USA, May 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.aisd-main.5. URL https://aclanthology.org/2025.aisd-main.5/.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI scientist: Towards fully automated open-ended scientific discovery, 2024. URL https://arxiv.org/abs/2408.06292. arXiv preprint, version 3.

Stefano Mammola, Diego Fontaneto, Alejandro Martínez, and Filipe Chichorro. Impact of the reference list features on the number of citations. *Scientometrics*, 126(1):785–799, 2021. doi: 10.1007/s11192-020-03759-0. URL https://doi.org/10.1007/s11192-020-03759-0.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279, 2022. URL https://aclanthology.org/2022.findings-acl.177/.

Minesh Mathew, Dimosthenis Karatzas, R. Manmatha, and C. V. Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2200–2209, 2021. URL https://arxiv.org/abs/2007.00398.

Shray Mathur, J. Anibal Boscoboinik, Esther H. R. Tsai, and Kevin G. Yager. LLM jaggedness unlocks scientific creativity, 2026. URL https://arxiv.org/abs/2605.10574.

Satyam Mukherjee, Daniel M. Romero, Ben Jones, and Brian Uzzi. The nearly universal link between the age of past knowledge and tomorrow's breakthroughs. *Science Advances*, 3(4):e1601315, 2017. doi: 10.1126/sciadv.1601315. URL https://doi.org/10.1126/sciadv.1601315.

Yansheng Qiu, Haoquan Zhang, Zhaopan Xu, Ming Li, Diping Song, Zheng Wang, and Kaipeng Zhang. AI idea bench 2025: AI research idea generation benchmark, 2025. URL https://arxiv.org/abs/2504.14191. arXiv preprint, version 3.

Kai Ruan, Xuan Wang, Jixiang Hong, Peng Wang, Yang Liu, and Hao Sun. Evaluating LLMs' divergent thinking capabilities for scientific idea generation with minimal context. *Nature Communications*, 17(3625), 2026. doi: 10.1038/s41467-026-70245-1. URL https://doi.org/10.1038/s41467-026-70245-1.

Yuanhao Shen, Daniel Xavier de Sousa, Ricardo Marçal, Hongyu Guo, and Xiaodan Zhu. IDRBench: Understanding the capability of large language models on interdisciplinary research, 2025. URL https://arxiv.org/abs/2507.15736. arXiv preprint; revised in 2026.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. Can LLMs generate novel research ideas? a large-scale human study with 100+ NLP researchers. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/ea94957d81b1c1caf87ef5319fa6b467-Abstract-Conference.html. ICLR 2025; arXiv:2409.04109.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024. URL https://arxiv.org/abs/2408.03314.

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. 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)*, pp. 28201–28240, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.1368. URL https://aclanthology.org/2025.acl-long.1368/.

Brian Uzzi, Satyam Mukherjee, Michael Stringer, and Ben Jones. Atypical combinations and scientific impact. *Science*, 342(6157):468–472, 2013. doi: 10.1126/science.1240474. URL https://doi.org/10.1126/science.1240474.

Rosni Vasu, Chandraye Basu, Bhavana Dalvi Mishra, Cristina Sarasua, Peter Clark, and Abraham Bernstein. Hyper: Literature-grounded hypothesis generation and distillation with provenance. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025. doi: 10.18653/v1/2025.emnlp-main.1292. URL https://aclanthology.org/2025.emnlp-main.1292/.

Jian Wang, Reinhilde Veugelers, and Paula Stephan. Bias against novelty in science: A cautionary tale for users of bibliometric indicators. *Research Policy*, 46(8):1416–1436, 2017. doi: 10.1016/j.respol.2017.06.006. URL https://doi.org/10.1016/j.respol.2017.06.006.

Wenxiao Wang, Lihui Gu, Liye Zhang, Yunxiang Luo, Yi Dai, Chen Shen, Liang Xie, Binbin Lin, Xiaofei He, and Jieping Ye. SciPIP: An LLM-based scientific paper idea proposer, 2025. URL https://arxiv.org/abs/2410.23166. arXiv preprint, version 2.

Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang, and Linyi Yang. CycleResearcher: Improving automated research via automated review, 2025. URL https://arxiv.org/abs/2411.00816. Accepted at ICLR 2025; version 3.

Yixuan Weng, Minjun Zhu, Qiujie Xie, Qiyao Sun, Zhen Lin, Sifan Liu, and Yue Zhang. DeepScientist: Advancing frontier-pushing scientific findings progressively. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.26603.

Yutaro Yamada et al. The AI scientist-v2: Workshop-level automated scientific discovery via agentic tree search, 2025. URL https://arxiv.org/abs/2504.08066.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Zonglin Yang, Wanhao Liu, Ben Gao, Tong Xie, Yuqiang Li, Wanli Ouyang, Soujanya Poria, Erik Cambria, and Dongzhan Zhou. MOOSE-Chem: Large language models for rediscovering unseen chemistry scientific hypotheses. In *International Conference on Learning Representations*, pp. 33251–33277, 2025. URL https://arxiv.org/abs/2410.07076.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qizhe Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3911–3921, 2018. doi: 10.18653/v1/D18-1425. URL https://aclanthology.org/D18-1425/.

Jiakang Yuan, Xiangchao Yan, Bo Zhang, Tao Chen, Botian Shi, Wanli Ouyang, Yu Qiao, Lei Bai, and Bowen Zhou. Dolphin: Moving towards closed-loop auto-research through thinking, practice, and feedback. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21768–21789, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.1056. URL https://aclanthology.org/2025.acl-long.1056/.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Sun, Zhen Wei, Guangnan Yu, Rushi Zhang, Zhi Song, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9556–9567, 2024. URL https://arxiv.org/abs/2311.16502.

Ming Zhang, Kexin Tan, Yueyuan Huang, Yujiong Shen, Chunchun Ma, Li Ju, Xinran Zhang, Yuhui Wang, Wenqing Jing, Jingyi Deng, and Huayu Sha. OpenNovelty: An LLM-powered agentic system for verifiable scholarly novelty assessment, 2026. URL https://arxiv.org/abs/2601.01576.

Keyu Zhao, Weiquan Lin, Qirui Zheng, Fengli Xu, and Yong Li. Deep ideation: Designing LLM agents to generate novel research ideas on scientific concept network, 2025. URL https://arxiv.org/abs/2511.02238.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Arjun I. Gonzalez, Daniel Fried, Uri Alon, and Dan Klein. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.13854.

A Extended Analysis of the Experimental Results

This appendix expands the analyses summarized in Section 4. No new numbers are introduced; all values refer to Tables 1, 2, and 3.

A.1 Controlled comparison scope

The controlled rows in Table 1 consume the benchmark-provided reference context and produce the benchmark-required textual idea under a fixed backbone. We do not mix paper-reported numbers from methods with a different interface. ResearchAgent requires a core paper, dynamically expanded citation neighbourhood, and external entity store (Baek et al., 2025); Chain of Ideas starts from a free-form topic and uses online retrieval and citation traversal (Li et al., 2025); and EIG evolves an internal proposal graph and reports only AI Idea Bench 2025 and LiveIdeaBench results (Dong et al., 2026). Reproducing these systems would require a non-unique adapter, a frozen retrieval snapshot, and an explicit information-access policy. Likewise, methods without an experimental-plan output are not applicable to plan-dependent metrics, following the task-specific handling of SciPIP in AI Idea Bench 2025 (Wang et al., 2025).

A.2 Generation: behaviour of each method family

Pure prompting is worst overall but not worst everywhere. Zero-shot, few-shot, and CoT occupy the bottom of almost every column: without external literature the model has nothing to ground a proposal in, so alignment-style metrics (Semantic Similarity, I2T/I2I, IDRBench composition) collapse, and IMCQ motivation drops to 0.107–0.147 against 0.2–0.3 for the graph methods. Their one relative strength is dispersion. Because nothing constrains what they may propose, their Novelty Insight (5.407–5.802) is not far below the graph-based methods and is clearly above our own Mainstream variant, and the benchmarks' native prompts—still pure prompting, but tuned per benchmark—reach the single best Novelty Insight of 6.823. Unconstrained sampling buys apparent novelty at the price of grounding; it does not buy quality.

Graph-based methods improve grounding uniformly and mildly. GoR, GraphMind, Deep Ideation, and GoAI all sit in a narrow band (Semantic Similarity 0.524–0.534, Overlap 6.5–7.1, composition 75.9–78.1), above prompting and RAG on nearly every metric. This is the expected effect of conditioning on literature structure: retrieved or graph-organized context makes proposals better anchored in existing work. These methods receive the same papers we do, but their representations organize citations, concept co-occurrence, or the proposal's own claim structure, so the experimental sections of those papers are not carried into the structure the generator conditions on. Nothing in their representation identifies which benchmark a proposal should be measured on. Their improvement is therefore uniform but shallow, and they never reach a best value on any metric.

Full-pipeline autonomous systems trade novelty for feasibility. Dolphin, AI-Scientist, and DeepScientist show a characteristic profile: the best Feasibility Insight among the baselines (6.423–6.483) and strong IMCQ experience (0.306–0.324), but the worst Novelty Insight in the entire table (3.847–4.028). These systems run an idea through implementation, review, and revision, and every revision step pushes a proposal toward what can actually be built and reviewed favourably. The result is proposals that are executable and well-motivated experimentally, but conservative.

The two SciGraph policies separate along the axis they weight. Our two policy rows reach best values on several grounding-type metrics, and the split follows directly from their weights. The Mainstream policy emphasizes the importance and validity priors and is associated with the best Semantic Similarity (0.542), Idea Overlap (7.454), Feasibility Insight (7.814), I2I motivation (3.906), and IDRBench composition (82.5). Its Novelty Insight is 4.062, among the lowest in the table. The Novelty policy emphasizes the novelty prior and is associated with Novelty Insight 6.643 (second best), the best I2T experience (4.024), I2I experience (3.884), and IMCQ experience (0.332), at the price of Feasibility Insight (6.184) and Overlap (6.535). These contrasting profiles arise from a controlled contrast on the same benchmark-derived candidate pool: the policies use the same graph construction and renderer but different selection criteria. They show the direction of the policy trade-off, not that either policy is superior to random selection.

Table 4: Generation quality under the same graph variants (Sem.: IdeaBench-research semantic similarity; I2T: AI Idea Bench 2025 motivation; Comp.: IDRBench composition).

Variant	Sem.↑	I2T↑	Comp.↑
Full graph	0.542	3.857	82.5
w/o citation	0.521	3.748	78.9
w/o protocol	0.517	3.790	77.6
Evidence shuffle	0.494	3.635	73.9
Edge shuffle	0.502	3.688	74.8
w/o mechanism	0.489	3.612	72.4
Zero-shot	0.455	3.427	65.7

A.3 Evaluation: why unstructured judging degenerates

All evaluators in this section receive the same input: the candidate text and its source papers. What follows is therefore about how that evidence is used, not about how much of it each method has.

LLM judges. On effectiveness, the four judges score 0.520–0.600 accuracy against a 0.54 majority-class rate, and their recall of 0.935–1.000 at precision 0.489–0.538 shows why: they call almost every candidate effective. A judge reading a fluent, well-motivated mechanism description does not go back through the experimental sections of its references to tally how comparable methods fared on the target protocol, and since every candidate reads plausibly, the decision degenerates to a near-constant. On ranking the instability is clearer still: the same four models produce novelty τ from -0.090 to $+0.200$, so the sign of the correlation changes with the backbone, and glm-5.2 attains the second-best novelty τ ($+0.200$) while simultaneously producing the worst importance correlation (-0.010, Top-1 $= 0.00$). A judge's agreement on one dimension therefore predicts nothing about its agreement on another. Uniformly high NDCG@5 (0.968–0.981) does not contradict this: with five candidates per group a poor ordering still retains most of the ideal gain, so NDCG@5 has little discriminative power in this setting and should be read together with τ and Top-1.

Benchmark-native evaluators. The native evaluators are prompt ensembles over the same backbones, and they behave accordingly. IdeaBench-research reaches novelty $\tau = +0.240$, the strongest baseline, plausibly because its Novelty Insight prompt explicitly asks how the proposal differs from the given references, which is a text-level proxy for combination rarity. LiveIdeaBench sits at $\tau = +0.010$, essentially no signal, and AI Idea Bench is negative at -0.030 on novelty while reaching only $+0.050$ on importance. Across the three, the spread on novelty is 0.27 in τ: the choice of evaluator changes the conclusion about which idea is more novel more than the ideas themselves do.

Graph-derived scoring. SciGraph attains 0.700 accuracy and 0.746 F1 on effectiveness, $\tau = +0.280/\rho = +0.340$/Top-1 $= 0.40$ on novelty, and $\tau = +0.300/\rho = +0.350$/Top-1 $= 0.30$ on importance— ranking first under all three reported dimensions. Each score is read from a different part of the graph rather than from one impression of the text. For effectiveness, the protocol view tabulates the outcomes of historical methods that used similar mechanisms on the same protocol, so a candidate whose mechanism neighbourhood has repeatedly failed on that benchmark is scored down even when its description is compelling; this is why the accuracy gain over the judges comes mainly from correctly rejecting candidates, which the unstructured judges never do. For novelty, the mechanism view counts the actual number of prior co-occurrences of each mechanism pair, so rarity is counted instead of guessed. For importance, the citation view records the citation standing of the papers carrying each mechanism, and the mechanism view measures how far those mechanisms sit from each other in the co-occurrence graph, so the score reflects what a combination is built on rather than how ambitiously it is phrased. The absolute levels remain modest, so the contribution is not a strong predictor but a scoring procedure whose every value decomposes into named nodes, edges, and paper rows Mechanism resolution is LLM-assisted; given the resolved mapping, the score is a deterministic graph calculation.

A.4 Ablations: view specialization and the content–structure asymmetry

The view-removal pattern matches how the scores are defined: mechanism co-occurrence drives rarity, citation standing drives grounding, and protocol history drives the effectiveness prior, so each removal degrades primarily the dimension it feeds. The structure controls separate two hypotheses. Text-shuffling keeps the typed edges but scrambles the evidence spans; degree-preserving shuffling keeps every node and every degree but breaks which node connects to which; random edges break the degree sequence as well. The monotone degradation across these three (text-shuffle $>$ degree-preserving $>$ random) indicates that the signal lives in the specific incidence structure, not merely in the presence of attached evidence.

On the generation side, corrupting the incidence structure costs 7.7–8.6 points of IDRBench composition. The corrupted variants retain roughly half of the full graph's margin over zero-shot prompting: 54% for edge shuffle and 49% for evidence shuffle. The asymmetry is the informative part: generation is helped mainly by graph *content*, which survives corruption, whereas scoring depends on graph *structure*, which does not. This is why the typed edges are reported as an evaluation-side contribution rather than as a generic prompting improvement.

B Prior Scores and External Reference Orders

All target orders used in the main evaluation are third-party reproducible reference orders rather than physical ground truth for unpublished ideas. We use real execution outcomes for effectiveness, the Wang-style first-combination fraction ($\mathrm{FracFirst}$) for novelty, and citation Grounding for importance. Both indicators come from the bibliometrics literature and are fixed before evaluating any judge; no LLM score, learned weight, or post-hoc tuning enters their definition, and no evaluator—including SciGraph—receives them as input.

B.1 Novelty reference order: Wang-style FracFirst

For a candidate or paper c with k mechanism mentions, we compute the Wang-style reference directly from the original-paper corpus available before its proposal or publication cutoff T. Let r_{ij}^{raw} be the raw-paper count for the pair of mentions (m_i, m_j). Following Wang et al. (Wang et al., 2017), we define

$$\mathrm{FracFirst}(c) = \frac{|\{(i,j) : r_{ij}^{\mathrm{raw}} = 0 \text{ in pre-}T\}|}{\binom{k}{2}}.$$

Thus, $\mathrm{FracFirst}(c) \in [0,1]$ is the fraction of pairs not observed in the pre-T original papers; larger values indicate a more novel combination. A pair involving an out-of-pool mechanism is treated as unobserved, rather than assigned a frequency or a similarity weight. Single-mechanism cases have no pair denominator and are recorded as missing for $\mathrm{FracFirst}$ ranking rather than assigned an arbitrary value.

This raw-paper reference is distinct from the SciGraph novelty prior below. $\mathrm{FracFirst}$ is computed directly from original-paper counts and is kept outside the graph pipeline. In contrast, Nov is computed from the weighted combination edges of the constructed graph after candidate-to-family resolution. The two quantities may be correlated because they address related novelty notions, but they are not the same count, are not derived from one another, and $\mathrm{FracFirst}$ is never an input to SciGraph.

This is the only novelty criterion used to construct the target ranking. Uzzi-style z-scores (Uzzi et al., 2013) and combination age (Mukherjee et al., 2017) were implemented as diagnostics during calibration, but are not used as reference orders. We use midranks for equal reference values and Kendall τ_b for tie-aware rank correlation. For Top-1, the reported protocol randomly breaks tied reference maxima over 10 repetitions and averages the resulting hit rate.

B.2 Importance reference order: citation Grounding

For a mechanism m, let $P_T(m)$ be the set of papers published before cutoff T that carry the mechanism. For each carrier paper p, $\mathrm{PR}_T(p)$ is its citation percentile within its same-year, same-venue cohort, computed using information available before T. Following the bibliometric grounding

view of Bornmann et al. (2010), with mean aggregation over carrier papers motivated by Mammola et al. (2021), define

$$\mathrm{CitationMass}_T(m) = \operatorname*{mean}_{p \in P_T(m)} \mathrm{PR}_T(p), \qquad \mathrm{Grounding}_T(c) = \operatorname*{median}_{m \in M(c)} \mathrm{CitationMass}_T(m).$$

Grounding is therefore high when the mechanisms used by an idea are supported by historically influential papers. If a mechanism has no auditable carrier papers or no valid citation percentile, its value is missing; missing values are never replaced by zero. Candidate Grounding is an ex-ante importance reference because an unpublished candidate has no citation history of its own.

This is the only importance criterion used to construct the target ranking. Protocol Anchor—the min–max position of a reported result within the historical results of the same protocol group, with direction reversed for lower-is-better metrics—and mechanism Between—Brandes betweenness centrality of a mechanism in the pre-cutoff task-level co-occurrence graph (Brandes, 2001)—are retained as exploratory graph features for prior scoring and ablations, but are not used as reference orders.

B.3 Ex-ante prior scoring and candidate selection

The three graph priors define the candidate-selection rule. For each SciGraph-Probe domain, we first generate one shared pool of $N = 30$ randomly sampled and rendered candidates. Mainstream and novelty-oriented then rank this same pool under their respective fixed policies. Each policy retains five candidates. If a candidate appears in both retained lists, it is assigned to the policy where it has the higher rank; for example, a candidate ranked second by novelty-oriented and third by mainstream is assigned to novelty-oriented. The other policy then retains its next highest-ranked unassigned candidate, so each policy contributes five candidates per domain and the two 50-candidate selections form the 100-candidate executed set. The policies do not generate the candidate text; they are fixed graph-conditional rules for ranking and selection. The same priors are also used by the SciGraph evaluator in Table 2: Val provides the effectiveness signal, while Nov and Imp provide the novelty and importance ranking signals after the candidate is resolved to the graph.

Table 1 is a separate generation experiment and does not use the 100 executed SciGraph-Probe candidates. Every compared system consumes the materials its benchmark provides, and we construct our graph from those same materials: the reference literature for IdeaBench-research, the target paper and inspiration materials for AI Idea Bench 2025, and the paper pair for IDRBench I3. Graph construction is thus an internal step applied to the shared input rather than an additional source of evidence. Each benchmark then receives the number of ideas required by its own protocol, and none of these benchmark ideas is executed.

For a candidate c, the mapping yields a mechanism set $M(c)$ and target protocol g, from which the three priors are read out of the graph. Mechanisms that cannot be resolved to a graph node are retained but marked out-of-graph; we write $M_{\mathrm{in}}(c)$ and $M_{\mathrm{out}}(c)$ for the in-graph and out-of-graph subsets. All statistics use only papers published before the candidate's proposal cutoff, and the candidate itself never enters the historical co-occurrence, citation, or protocol statistics.

Novelty prior (mechanism view). After the candidate is resolved to canonical mechanism families, let w_{ij}^G be the weight of the corresponding combination edge in the constructed graph, and let w_{ref}^G be the largest valid combination-edge weight in that graph view. Pair rarity and the out-of-graph ratio are

$$R_{\mathrm{pair}}(c) = 1 - \frac{1}{\binom{k}{2}} \sum_{i<j} \min\Big(1, \frac{w_{ij}^G}{w_{\mathrm{ref}}^G}\Big), \qquad R_{\mathrm{out}}(c) = \frac{|M_{\mathrm{out}}(c)|}{|M(c)|}, \qquad \mathrm{Nov}(c) = \alpha R_{\mathrm{pair}}(c) + (1-\alpha) R_{\mathrm{out}}(c),$$

with $k = |M(c)|$ and pairs having no graph combination edge contributing $w_{ij}^G = 0$. We set $\alpha = 0.8$, so rarity of observed combinations dominates and unresolved mechanisms contribute a minority share; this prevents extraction failures from being rewarded as novelty. We define $\mathrm{Coverage}(c) = |M_{\mathrm{in}}(c)|/|M(c)|$ and never treat low coverage as low importance or low validity—it only signals insufficient historical evidence.

Importance prior (mechanism + citation views). Two components are combined. Bridging uses Brandes betweenness centrality (Brandes, 2001) on the historical mechanism co-occurrence graph,

normalized within the task,

$$B_{\text{norm}}(m) = \frac{B(m)}{\max_{m'} B(m')}, \qquad I_{\text{bridge}}(c) = \underset{m \in M_{\text{in}}(c)}{\text{mean}} B_{\text{norm}}(m),$$

which is high when the candidate's mechanisms lie on many shortest paths between otherwise separate research lines. Grounding uses the citation standing of the papers that carry each mechanism. With $\text{Cit}(p) = \log(1 + \text{indeg}(p))/\log(1 + \max_q \text{indeg}(q))$ the normalized in-pool citation count and $P(m)$ the set of historical carrier papers of m,

$$I_{\text{ground}}(m) = \underset{p \in P(m)}{\text{mean}} \text{Cit}(p), \qquad I_{\text{ground}}(c) = \underset{m \in M_{\text{in}}(c)}{\text{median}} I_{\text{ground}}(m), \qquad \text{Imp}(c) = \beta_1 I_{\text{bridge}}(c) + \beta_2 I_{\text{ground}}(c),$$

with $\beta_1 + \beta_2 = 1$. We set $\beta_1 = 0.4$ and $\beta_2 = 0.6$. The mean is taken over carrier papers and the median across mechanisms, so that a single highly cited classic cannot dominate a candidate's score; a mechanism with no auditable carrier paper is missing rather than zero. Note that the prior uses in-pool normalized citation counts, whereas the importance reference order of Appendix B uses same-year, same-venue citation percentiles computed under strict pre-cutoff filtering—the prior is a cheap candidate-evaluation signal, not the reference order.

Validity prior (mechanism + protocol views). The validity prior does not infer effectiveness from popularity; it asks how often historical methods with similar mechanisms beat their baseline on the *same* protocol. For a historical method p reporting protocol g, the direction-normalized gap and its outcome label are

$$\Delta(p, g) = \begin{cases} \text{method}(p, g) - \text{baseline}(p, g), & \text{higher-is-better}, \\ \text{baseline}(p, g) - \text{method}(p, g), & \text{lower-is-better}, \end{cases} \qquad y(p, g) = \mathbb{1}[\Delta(p, g) > 0].$$

With Jaccard mechanism similarity $\text{sim}(c, p) = |M(c) \cap M(p)|/|M(c) \cup M(p)|$ and the neighbourhood $H(c, g) = \{p : \ p \text{ reports } g, \ \text{sim}(c, p) \geq \tau\}$, the prior is a similarity-weighted historical success rate smoothed toward the protocol's overall success rate $\bar{y}(g)$:

$$\text{Val}(c, g) = \frac{\sum_{p \in H(c,g)} \text{sim}(c, p)\, y(p, g) + \kappa\, \bar{y}(g)}{\sum_{p \in H(c,g)} \text{sim}(c, p) + \kappa}.$$

We set the similarity floor $\tau = 0.2$ and the smoothing strength $\kappa = 3$, so that a candidate matched by only one or two historical methods is pulled toward the protocol prior rather than taking an extreme value. Candidates targeting several protocols average over them with equal weight. Thus, $\text{Val}(c, g)$ is the continuous graph-derived validity prior used in policy selection and as SciGraph's effectiveness signal in Table 2; $\text{Nov}(c)$ and $\text{Imp}(c)$ are its novelty and importance ranking signals. For Table 2, a candidate with defined $\text{Val}(c, g)$ is classified as effective when $\text{Val}(c, g) \geq 0.5$ and ineffective otherwise. The threshold is fixed by the interpretation of Val as a smoothed historical success rate and is not tuned on the 46/54 execution labels. If a protocol group has no historical baseline comparison, Val is reported as unknown and is not substituted by a default probability, by mechanism centrality, or by citation influence.

Policy weights. All three priors lie in $[0, 1]$ and are min–max normalized within a task before weighting. The two policies differ only in w^π:

$$w_{\text{main}} = (w_{\text{nov}}, w_{\text{imp}}, w_{\text{val}}) = (0.2,\ 0.4,\ 0.4), \qquad w_{\text{novel}} = (0.6,\ 0.2,\ 0.2).$$

For either policy π, the selection score is the weighted sum of the three priors, $S^\pi(c) = w^\pi_{\text{nov}} \text{Nov}(c) + w^\pi_{\text{imp}} \text{Imp}(c) + w^\pi_{\text{val}} \text{Val}(c, g)$. The policy ranks generated candidates by this score and selects the required outputs. The resulting selection decision is traceable to mechanisms, papers, and protocol records. The weights are fixed before execution, are not learned from execution outcomes, and are not used to define any reference order. The external reference orders in Appendix B remain outside this pipeline and are used only to evaluate the evaluators. We do not claim that selection by these policies is superior to random selection, because candidates excluded by the policies were not executed.

B.4 Calibration on 184 real papers

We calibrate the two selected indicators on the 184 real papers in the ten-domain corpus. For each paper, all ex-ante quantities use only papers, mechanisms, and citation events available before its

Table 5: Novelty calibration on the 184-paper corpus. The selected $\mathrm{FracFirst}$ signal is compared with exploratory alternatives; the latter are not used as reference orders.

Window	Signal	n	Spearman ρ	Kendall τ_b
1 year	$\mathrm{FracFirst}$	153	0.4629	0.3521
1 year	Uzzi $-z$ (diagnostic)	104	0.1472	0.1022
1 year	Composition age (diagnostic)	85	-0.0843	-0.0690
2 years	$\mathrm{FracFirst}$	97	0.5179	0.4151
2 years	Uzzi $-z$ (diagnostic)	51	0.1490	0.1146
2 years	Composition age (diagnostic)	38	0.3539	0.2960

Table 6: Importance calibration on the 184-paper corpus. The table reports the selected Grounding signal; exploratory Anchor and Between results are summarized in the text and are not used as reference orders.

Window	Sample	n	Spearman ρ	Kendall τ_b	Top-1
1 year	Complete	127	0.2906	0.2227	Hit
1 year	Strict, no unknown years	52	0.3345	0.2662	Hit
2 years	Complete	74	0.2612	0.2086	Hit
2 years	Strict, no unknown years	14	0.4482	0.3822	Hit

first publication date: carrier papers are restricted to earlier publications, Grounding counts only citation events occurring before the focal date, Anchor uses only earlier protocol results, and the focal paper never enters its own reference set. Citation retrieval succeeded for 178 of 184 papers; after requiring a complete one-year window and a same-year, same-venue comparison cohort, 133 papers remained for the paper-level ex-post analysis. The ex-post target is each paper's citation percentile within its same-year, same-venue cohort over a complete one- or two-year post-publication window. We report Spearman ρ and Kendall τ_b against this subsequent citation percentile. The sample size varies because incomplete windows, missing venues, and missing mechanism evidence are retained as missing rather than imputed; citation events with unknown year are excluded from exact windows rather than counted as zero, so the "complete" samples treat window citation counts as lower bounds and the "strict" samples exclude such papers entirely.

$\mathrm{FracFirst}$ is the most consistent novelty signal in this corpus: it has the strongest positive association in both the one- and two-year windows, while the alternatives are weaker or unstable and have lower coverage. This supports using $\mathrm{FracFirst}$ as the novelty reference order, without claiming that the correlation establishes a universal causal law about scientific impact.

Grounding has a positive association with ex-post citation percentile in every reported window and remains positive when unknown-year citation events are excluded. In the same calibration, Anchor is unstable across windows ($\rho = 0.1529$ at one year but $\rho = -0.1529$ at two years, with only 51 and 23 valid papers respectively) and Between is near zero or negative in the principal samples ($\rho = -0.0512$ at one year, $\rho = -0.1121$ at two years); we therefore do not use either as a reference order, although both are retained as prior-scoring features where a weak or unverified signal is acceptable. As a threshold-free summary we also report the share of papers whose ex-post citation percentile reaches or exceeds the mid-rank percentile of their ex-ante score: 82.7% $(110/133)$ for $\mathrm{FracFirst}$ and 78.0% $(99/127)$ for Grounding at one year, against 71.1% for Uzzi $-z$, 74.4% for combination age, 80.4% for Anchor ($n = 51$), and 74.0% for Between. These are one-sided hit rates, not classification accuracies. Overall the Grounding and $\mathrm{FracFirst}$ results provide preliminary corpus-level calibration rather than a claim of universal validity, especially because the strict two-year sample contains only 14 papers, the corpus is a task-filtered AI pool, and citation events outside the pool are incomplete so window counts are lower bounds.

B.5 Dataset composition and candidate inventory

We provide the composition of the 184-paper corpus and a small set of execution records selected for qualitative inspection. The task domains, representative paper sources, and frequently used benchmarks are summarized in Table 7. Appendix B.6 reports five mixed-outcome records with their

Table 7: Representative paper sources and frequently used benchmarks in each task domain. The benchmark list is broader than the candidate-specific protocols; the corpus contains 184 papers and 100 candidates in total.

ID	Task domain	Representative paper sources	Frequently used benchmarks
T1	Reasoning, verification, and test-time scaling	ICLR 2022/2024; NeurIPS 2023–2024; KDD 2023; ACL 2025; arXiv 2024	GSM8K (Cobbe et al., 2021); MATH; GPQA; AIME24; MMLU-Pro (Hendrycks et al., 2021)
T2	Retrieval-augmented generation and citation grounding	ICLR 2023; NeurIPS 2024; SIGIR 2024; ACL 2025; arXiv 2024–2026	KILT; HotpotQA (Yang et al., 2018); MuSiQue; RGB; RAGChecker
T3	Long-context understanding	ACL/EMNLP 2023–2024; ICLR 2024; ICML 2024; TACL 2023; arXiv 2023–2024	LongBench (Bai et al., 2024); LongBench v2; RULER; InfiniteBench; NeedleBench; L-Eval
T4	Code generation with execution feedback	ICLR 2024; NeurIPS 2023–2024; ACL 2024–2025; arXiv 2023–2025	HumanEval+ (Chen et al., 2021); MBPP; APPS; LiveCodeBench; BigCodeBench
T5	Repository-level issue resolution	NeurIPS 2024–2025; ASE 2025; ISSTA 2024; ACM Software Engineering 2024; arXiv 2023–2024	SWE-bench (Jimenez et al., 2024); SWE-bench Verified; Mini-SWE-bench; SWE-Lancer; Defects4J
T6	GUI agents	ACL 2024/2026; NeurIPS 2023–2024; SIGIR 2026; arXiv 2022–2024	WebArena (Zhou et al., 2024); VisualWebArena; OSWorld; AppWorld; Mind2Web
T7	Tool use and API calling	NeurIPS 2023–2024; ACL/NAACL 2024–2025; arXiv 2023–2025	BFCL; ToolSandbox; τ-bench; API-Bank (Li et al., 2023); ToolTalk
T8	Text-to-SQL	AAAI 2023/2025; NeurIPS 2023; ACL 2026; IEEE 2023–2025; arXiv 2023–2024	BIRD; Spider 2.0; Spider (Yu et al., 2018); WikiSQL; LiveSQLBench
T9	Document and chart understanding	CVPR/ICCV 2023–2025; ACL 2022–2025; TPAMI; LNCS 2024	DocVQA (Mathew et al., 2021); ChartQA (Masry et al., 2022); OCR-Bench; ScreenSpot; InfoVQA
T10	Multimodal reasoning	NeurIPS 2025; CVPR 2023/2025; ACL 2025–2026; AAAI 2025; EMNLP 2024; arXiv 2023	MMMU (Yue et al., 2024); MathVista; MMBench; MMVet; HallusionBench

Table 8: Execution outcomes by candidate type. The table compares mainstream and novelty candidates only.

Candidate type	Total	Effective	Non-effective	Effective rate
Mainstream	50	26	24	52%
Novelty	50	20	30	40%

candidate mechanisms, fixed baseline setting, observed metric change, and recorded execution note. The examples are selected for interpretability and coverage; they are not a substitute for aggregate reporting over all 100 candidates.

Candidate type and execution outcome. Table 8 compares the two endpoint candidate types. Novelty candidates have a slightly lower effective rate than mainstream candidates. One plausible explanation is that a novelty-oriented mechanism combination has less direct historical precedent on the target protocol, which increases uncertainty during implementation and execution. This is an observational pattern rather than a causal result: the types differ in more than novelty alone, and the table does not isolate the effect of any single mechanism or selection decision.

Table 9: Traceable graph-to-execution case studies. Each row links a candidate to resolved graph evidence, then to a fixed protocol comparison. M/P refs are resolved mechanism/protocol records; Δ is candidate minus baseline. The last column summarizes the recorded outcome only.

Case	Candidate and graph trace	Fixed protocol comparison	Observed outcome	What the case makes inspectable
T3-F	**Hierarchical Summarization**: long-context alignment + memory; 6/3 M/P refs	InfiniteBench longbook_qa; truncated-context QA; accuracy	$0.20 \rightarrow 0.12$ ($\Delta = -0.08$), ineffective	Summary-based compression loses answer-critical details in this recorded long-context setting.
T6-H	**Plan-Execution Memory**: browser planning + memory; 8/9 M/P refs	OSWorld; single action-plan generation; plan success	$0.825 \rightarrow 0.750$ ($\Delta = -0.075$), ineffective	Reusing historical plans does not transfer reliably to this GUI setting.
T7-G	**Execution-Verified Selection**: tool selection + execution feedback; 3/1 M/P refs	NESTFUL; single function-call generation; accuracy	$0.7933 \rightarrow 0.8867$ ($\Delta = +0.0934$), effective	Execution verification and recovery coincide with a positive outcome on nested tool calling.
T8-H	**Decomposition and Assembly**: SQL decomposition + CoT; 5/15 M/P refs	BIRD; single SQL generation; semantic equivalence	$0.17 \rightarrow 0.23$ ($\Delta = +0.06$), effective	Query decomposition is associated with a positive recorded change in this Text-to-SQL protocol.
T10-I	**Hallucination-Checking Reasoning**: hallucination mitigation + CoT; 7/5 M/P refs	MathVerse; single-pass VLM answer; accuracy	$0.94 \rightarrow 0.91$ ($\Delta = -0.03$), ineffective	An added checking stage can interfere with a fixed multimodal reasoning protocol.

Table 10: T2/RGB rank-perturbation case study. Candidate content is unchanged between the canonical and rewritten LLM prompts.

Property	Method	Rank (high $\rightarrow$ low)	Spearman ρ	Kendall τ_b	Paired agreement
Novelty	LLM canonical	C > B > E > A > D	+0.112	+0.120	rewrite: $\rho = 0.900$, $\tau = 0.800$
Novelty	LLM rewrite	C > E > B > A > D	+0.335	+0.359	canonical: $\rho = 0.900$, $\tau = 0.800$
Novelty	SciGraph	C > D > A > B > E	+0.224	+0.120	—
Importance	LLM canonical	A > C > D > E > B	+0.051	+0.105	rewrite: $\rho = 0.900$, $\tau = 0.800$
Importance	LLM rewrite	A > C > E > D > B	-0.103	-0.105	canonical: $\rho = 0.900$, $\tau = 0.800$
Importance	SciGraph	D > A > E > C > B	+0.872	+0.738	—

B.6 TRACEABLE GRAPH-TO-EXECUTION CASE STUDIES

Table 9 follows five candidates from graph evidence to execution. We select cases with resolved mechanisms, at least one protocol reference, nontrivial absolute change, and no stated protocol or implementation caveat; we exclude near-tie outcomes and unusually large gains. For each case, "M/P refs" gives the number of resolved mechanism and protocol records attached to the candidate before execution. The final column is a bounded interpretation of the recorded outcome, not a claim of repeatability, statistical significance, or causal attribution. This qualitative selection is not used in any aggregate metric, reference order, or statistical test.

Cross-case reading. The two positive and three negative cases show that the same graph can expose candidate mechanisms and protocol evidence while leaving the execution outcome open. This is the intended audit trail: a reader can inspect the candidate, its resolved evidence, the fixed comparison, and the observed result without treating the trace as causal proof or reducing novelty, importance, and effectiveness to a single score.

Rank perturbation in T2/RGB. Table 10 uses a fixed five-candidate T2/RGB citation-F1 group. Its external novelty reference order is computed by tie-aware $\mathrm{FracFirst}$: C, D, and E are tied first, followed by B and A. Its external importance reference order is Grounding: D is first, A and E are tied next, followed by B and C. Neither reference order is provided to the evaluators.

We compare a canonical LLM prompt with a semantically equivalent rewrite. The rewrite changes only the wording and presentation order of the instruction; it preserves the candidate information, requested property, ranking task, and output format.

The LLM ranking changes under the equivalent rewrite: its agreement with the canonical prompt is $\rho = 0.900$ and $\tau = 0.800$ for both properties. This prompt-dependent variation makes the judgment less reliable as a fixed ranking procedure.

For novelty, the rewritten LLM prompt is closer to the reference order ($\rho = 0.335$, $\tau_b = 0.359$) than SciGraph ($\rho = 0.224$, $\tau_b = 0.120$). SciGraph is therefore not uniformly better in every group. For importance, however, SciGraph has much stronger agreement with Grounding ($\rho = 0.872$, $\tau_b = 0.738$) than either LLM prompt, whose agreement is near zero or negative. This case illustrates that graph-based scoring can be more aligned with the importance reference order in this group, while LLM judgments can occasionally be closer to a reference order.