Intern-Atlas: A Methodological Evolution Graph as Research Infrastructure for AI Scientists

Yujun Wu[1,2*] **Dongxu Zhang**[3,4,2*] **Xinchen Li**[5,2*] **Yiling Duan**[6*] **Jinhang Xu**[7,2*]
Yumou Liu[8,2] **Jiabao Pan**[9,2] **Qiyuan Zhu**[2] **Xuanhe Zhou**[8] **Jingxuan Wei**[10]
Siyuan Li[5] **Jintao Chen**[5] **Conghui He**[2] **Cheng Tan**[2†]
[1]Shenzhen Graduate School, Peking University
[2]Shanghai Artificial Intelligence Laboratory
[3]School of Software Engineering, Xi'an Jiaotong University
[4]Department of Precision Instrument, Tsinghua University
[5]Zhejiang University
[6]National Graduate College for Elite Engineers, Hunan University
[7]East China Normal University [8]Shanghai Jiao Tong University
[9]Shanghai University [10]University of Chinese Academy of Sciences

Abstract

Existing research infrastructure is fundamentally document-centric, providing citation links between papers but lacking explicit representations of methodological evolution. In particular, it does not capture the structured relationships that explain how and why research methods emerge, adapt, and build upon one another. With the rise of AI-driven research agents as a new class of consumers of scientific knowledge, this limitation becomes increasingly consequential, as such agents cannot reliably reconstruct method evolution topologies from unstructured text. We introduce Intern-Atlas, a methodological evolution graph that automatically identifies method-level entities, infers lineage relationships among methodologies, and captures the bottlenecks that drive transitions between successive innovations. Built from $1{,}030{,}314$ papers spanning six decades of AI research from 1965 onward, covering conferences, journals, and arXiv preprints, the resulting graph comprises $9{,}410{,}201$ semantically typed edges, forming a queryable causal network of methodological development. We adapt self-guided temporal tree search to retrieve chains of methods from this graph. We evaluate the quality of the resulting graph against expert-curated ground-truth evolution chains and observe strong alignment. In addition, we demonstrate that Intern-Atlas enables downstream applications in idea evaluation and automated idea generation. We position methodological evolution graphs as a foundational data layer for the emerging field of automated scientific discovery.

1 Introduction

Scientific progress in artificial intelligence has traditionally been documented through papers, and the infrastructure developed to organize this body of knowledge reflects a document-centric paradigm. Platforms such as Google Scholar [1], Semantic Scholar [2], and OpenAlex [3] treat the paper as the atomic unit: they index titles, abstracts, and citation counts, and they connect papers to one another through citation links. This paradigm has served human researchers for decades. To understand the emergence of vision transformers, for example, a researcher can retrieve a set of relevant papers from

*These authors contributed equally and share first authorship.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

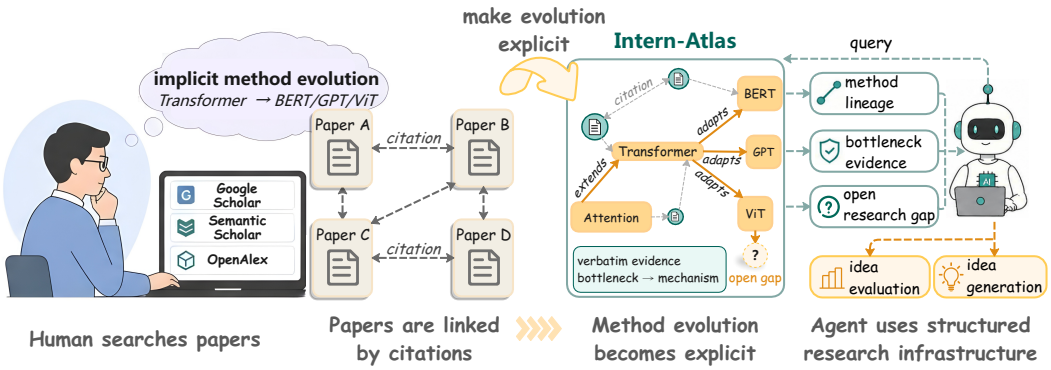


Figure 1: **From paper lists to an agent-readable methodology atlas.** Citation-based discovery tools (left) leave *method* evolution (*Transformer* → *BERT*/*GPT*/*ViT*) implicit in human expertise. INTERN-ATLAS (middle) makes it explicit as a typed graph with causal edge labels and verbatim bottleneck-to-mechanism evidence, providing an LLM agent (right) with direct queries for method lineage, bottleneck evidence, idea evaluation, and idea generation.

these systems, read them, and reconstruct the lineage from convolutional neural networks through self-attention mechanisms to modern architectures. The critical step in this workflow, extracting the structural relationships among methods from narrative text and assembling them into a coherent evolutionary picture, is performed entirely inside the researcher's head (Figure 1, left).

A new class of knowledge consumer is now emerging. AI-driven research agents that automate the scientific pipeline from hypothesis generation to experimentation [4, 5, 6] represent a fundamentally different kind of knowledge consumer. Unlike humans, these agents cannot reliably reconstruct method evolution topologies from unstructured text. Their parametric memory is a lossy compression that underrepresents low-frequency or long-tail methodological knowledge. Their autoregressive inference operates as fixed-depth forward computation rather than explicit graph traversal, limiting their ability to enumerate branching method spaces. Most critically, they lack the capacity to distinguish between genuine gaps in the research landscape and gaps in their own internal representations, since both manifest as an absence of relevant activations. As a result, their most persistent limitation lies in idea generation, where output quality depends on structural understanding of the methodological landscape: not only which methods exist, but how they evolved, what constraints they addressed, and which directions remain unexplored.

This gap reflects a recurring historical pattern. Structured knowledge infrastructure becomes essential not when humans require it, but when automated systems emerge that cannot operate effectively over unstructured data. The Protein Data Bank (PDB) [7] standardized protein structures decades before AlphaFold [8] revealed their full value as machine-consumable training data. ImageNet [9] organized visual data into hierarchical labels prior to the widespread adoption of convolutional neural networks that depended on large-scale structured annotations. In each case, the emergence of a new computational consumer transformed latent structure into an explicit requirement. A similar inflection point is now occurring in scientific methodology: *AI research agents have emerged, yet the structured data layer required to support them remains absent*.

We introduce Intern-Atlas, a methodological evolution graph that fills this infrastructure gap. Atlas processes papers from top-tier AI venues, extracts method entities with alias resolution, classifies every citation edge into semantic types, and grounds each non-background edge in a verbatim quote with structured bottleneck and mechanism annotations. The resulting graph forms a queryable causal topology of methodological evolution, providing a persistent data layer that can be directly consumed by downstream systems (Figure 1, right). Beyond graph construction, identifying meaningful evolution chains introduces an additional challenge. Methodological progress forms a directed acyclic graph: a single method such as the Transformer [10] gives rise to multiple branches, including BERT [11], GPT [12, 13, 14], Vision Transformer [15, 16, 17], and DETR [18]. Greedy traversal strategies that commit at each branching point discard alternative trajectories. We adapt Self-Guided

Temporal Monte Carlo Tree Search (SGT-MCTS) to the method graph, balancing high-confidence paths with less explored branches while respecting publication order.

We evaluate Intern-Atlas along three complementary dimensions. We first assess intrinsic graph quality by quantitatively comparing constructed evolution chains against expert-curated chains. We then examine its utility for idea evaluation, testing whether graph-grounded scoring signals exhibit monotonic alignment with human tiers. Finally, we investigate its value for idea generation by comparing outputs produced with and without access to graph-derived evolutionary context. Taken together, our results suggest that methodological evolution graphs can serve as a foundational data layer for the emerging ecosystem of automated scientific discovery.

2 Related Work

2.1 Scientific Knowledge Graphs

Tracing knowledge flow through citation networks has a long history. Main Path Analysis [19] traces routes through direct citation links using traversal counts. CiteSpace [20] detects research fronts through keyword burst analysis and co-citation clustering. Modern large-scale platforms have significantly expanded the metadata available for scientific literature. OpenAlex [3] indexes hundreds of millions of works; Semantic Scholar [2] and S2ORC [21] augment standard citation graphs with contextual sentences and influential-citation markers; and Papers With Code introduces structured Task-Dataset-Metric triples. A paper-level citation alone leaves open whether the citing paper extends an earlier architecture, replaces a component, or mentions the earlier work as background.

THE-Tree [22] starts from surveys of a topic and uses SGT-MCTS to expand a tree of papers, checking proposed links against the cited literature. The tree supports scientific verification and prediction of later developments. Intern-Atlas uses a two-pass LLM extractor on citation contexts across its corpus and maps method aliases to canonical nodes. SGT-MCTS then retrieves lineages from this precomputed method graph, which also supplies evidence for idea evaluation and generation.

2.2 Evaluating Research Ideas

Experts can disagree when judging the novelty of a research idea [23], and independent reviews of the same paper can reach different conclusions [24]. AI Idea Bench [25] compares proposed ideas with earlier and contemporary work. IdeaBench [26] ranks proposals on criteria such as novelty and feasibility, while the Ideation–Execution Gap study [27] follows ideas through implementation and expert review.

SciMuse [28] uses a graph of scientific concepts to generate ideas for expert evaluation. Intern-Atlas uses a graph of methods in the evaluation itself. For example, one component of its novelty score measures how closely the proposed methods are connected through their use in earlier papers (Section 3.3.2).

2.3 LLM-based Scientific Ideation

End-to-end automated research systems have rapidly evolved from template-driven pipelines to sophisticated multi-agent loops. Early iterations like AI Scientist v1 [4] demonstrated fully automated workflows but relied heavily on hand-crafted templates. Subsequent architectures have introduced dynamic problem-solving: AI Scientist v2 [29] utilizes agentic tree search to generate workshop-accepted papers; CycleResearcher [30] closes the research-review-revision loop to reach preprint-level quality; Dolphin [31] integrates exception-traceback-guided debugging; and AIGS [32] explicitly embeds falsification principles into the research process. Parallel to execution, automated idea generation has seen significant focus. Chain of Ideas (CoI) [33] structures literature into linear chains for LLM-guided extrapolation, while systems like Nova [34] and SciMON [35] optimize iterative search for novelty. However, both ideation and execution agents share a critical structural bottleneck: they construct their knowledge representations from scratch at task launch. CoI assembles chains transiently within prompts; AI Scientist hardcodes knowledge into templates; and SPARK [5] re-retrieves unstructured text per query. This repeated, transient reinvention across independent systems constitutes explicit behavioral evidence of a missing infrastructure layer. Intern-Atlas fills this void by

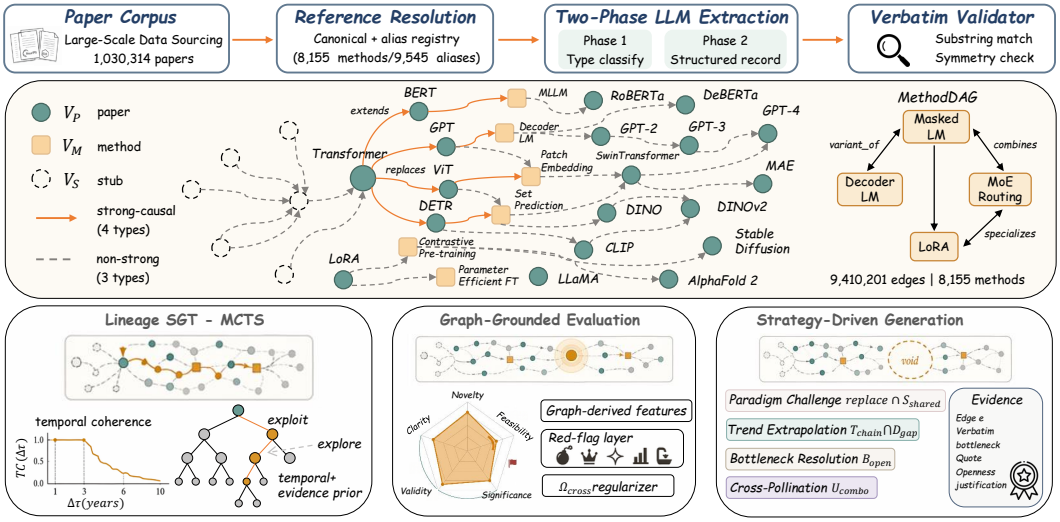


Figure 2: **Overview of INTERN-ATLAS. Top:** From 1,030,314 scholarly publications spanning conferences, journals, and arXiv preprints, we construct a typed methodology graph $G = (\mathcal{V}, \mathcal{E}, \tau, \rho)$ via reference resolution into papers, canonical methods, and stubs ($\mathcal{V}_P/\mathcal{V}_M/\mathcal{V}_S$; 8,155 methods, 9,545 aliases), two-phase LLM extraction (type classification $\rightarrow$ structured record), and a code-only verbatim validator. **Middle:** the resulting graph contains 9,410,201 typed edges over 8,155 canonical methods; strong-causal edges (4 types, solid) induce the lineage subgraph G_{strong}, non-strong edges (3 types, dashed) supply retrieval context, and the projected method-level DAG G_M. **Bottom:** *Lineage SGT-MCTS* reconstruction with an evidence-physics prior over edge confidence and temporal coherence $TC(\Delta\tau)$; *graph-grounded idea evaluation* on five axes with a parameter-free core function, red-flag detectors, and the cross-dimensional regularizer Ω_{cross}; and *strategy-driven idea generation* along four topological strategies, each proposal certified by a verbatim evidence record.

serving as their foundational prerequisite, providing the persistent, queryable, and method-granular knowledge base that current agents fundamentally lack.

3 Method

3.1 Overview

Existing scholarly platforms [2, 3] treat the paper as the atomic unit and connect papers through citations that carry no semantic information. For an AI research agent, this representation is impoverished: *it cannot tell whether paper A extends paper B's architecture, addresses B's specific bottleneck with a different mechanism, or merely mentions B as background.* We introduce **Intern-Atlas**, a method-centric heterogeneous graph G that closes this gap.

The system (Figure 2) operates in two stages. The first stage, **Methodological Graph Construction** (§3.2), turns raw AI papers into G by typing every citation edge with one of seven relations and attaching a four-field evidence record (bottleneck, mechanism, trade-off, confidence) to each causal edge. The second stage, **Operators over the Graph** (§3.3), instantiates three reasoning tasks on top of G: **lineage reconstruction** traces the causal ancestors and descendants of a target method through G; **idea evaluation** scores an idea by where its methods sit in G; and **idea generation** searches G for under-explored regions and proposes new ideas to occupy them.

3.2 Methodological Graph Construction

We build G in three steps from a corpus of 1,030,314 papers spanning 1965 to early 2026.

Step 1: entity resolution. The graph has three node sets: papers $\mathcal{V}_P$ (the parsed corpus), methods $\mathcal{V}_M$ (one node per method), and stubs $\mathcal{V}_S$ (placeholders for cited works that fall outside the corpus, totaling 3,173,187 nodes). $\mathcal{V}_M$ is seeded from a hand-curated list of well-known methods and

expanded by an LLM proposer for additional candidates. Because the same method appears under different names in raw text (e.g., Transformer vs. vanilla Transformer), we collect these names into a lookup table $\mathcal{A}$: string $\to \mathcal{V}_M$ that routes every observed name to the matching method node.

Step 2: edge typing. Each resolved reference becomes a typed edge $e = (u, v)$ from a method node $u \in \mathcal{V}_M$ to its target $v \in \mathcal{V}_M \cup \mathcal{V}_S$. An LLM classifier reads the citing paper's text near the reference and assigns e one of seven labels, ordered by decreasing causal strength: *extends*, *improves*, *replaces*, *adapts*, *uses_component*, *compares*, and *background*. We write type(e) for this label. The first four categories denote a direct methodological lineage and form the *strong-causal* subset $\mathcal{E}_{\text{sc}} \subseteq E$, on which lineage reconstruction operates; the remaining three are retained for retrieval context. Edges with type$(e) \neq$ *background* are called *causal edges* and proceed to Step 3.

Step 3: evidence extraction. For every causal edge, an LLM extractor populates a four-field evidence record:

$$\rho(e) \;=\; \big(\, b_e,\; m_e,\; t_e,\; c_e \,\big), \tag{1}$$

where b_e is the bottleneck addressed, m_e is the mechanism, t_e is the trade-off (each a text span quoted from the citing paper), and $c_e \in [0, 1]$ is an LLM-reported confidence. Each bottleneck is also tagged with one of 14 categories from a taxonomy $\mathcal{D}$ (e.g., *computational cost*, *long-range dependency*). This grouping lets the idea generator reason about which problem categories still lack a strong solution, rather than treating every bottleneck quote as unique. A deterministic post-checker discards edges that fail substring match, violate year ordering, or duplicate an opposite-direction edge. Further details are in Appendix A.

3.3 Operators over the Methodological Graph

All three operators begin the same way. Given a textual input x (a target method, an idea description, or a research question), each operator first uses the lookup table $\mathcal{A}$ to find method nodes matching x, then expands outward through BM25 [36] keyword retrieval over $\mathcal{V}_P$ to collect related papers and the typed edges among them. We denote the resulting localized context by:

$$\mathcal{C}_x \;=\; \text{Retrieve}(x, G, \mathcal{A}). \tag{2}$$

3.3.1 Lineage Reconstruction with SGT-MCTS

Lineage reconstruction unfolds a method's evolutionary history. Starting from each seed in $\mathcal{S}(q) \subseteq \mathcal{C}_q$ that matches the query q exactly, it returns a small set of evolution chains Π_q, where each $\pi \in \Pi_q$ is a directed path through the strong causal subgraph $(\mathcal{V}_M, \mathcal{E}_{\text{sc}})$ respecting publication-year order. Following THE-Tree [22], we use temporally guided tree search to explore alternative successors. Our UCT [37] selection rule uses the relation metadata already stored in G. At node u, child v is scored by:

$$\text{SGT-UCT}(v) \;=\; \text{UCT}(u, v) \;+\; \lambda \cdot \alpha_G(u, v), \tag{3}$$

where $\text{UCT}(u, v)$ is the standard exploit-explore criterion and $\alpha_G(u, v) = \text{conf}(e_{u \to v}) \cdot \text{TC}(\Delta\tau_{uv})$ multiplies the LLM-reported edge confidence (stored in $\rho(e_{u \to v})$, $\in [0, 1]$) by a temporal-coherence term peaked at year gaps $\Delta\tau \in [1, 3]$. Backward and forward chains from each seed are joined and ranked by:

$$\text{rank}(\pi) \;=\; w_\ell \, \tfrac{|\pi|}{L_{\max}} + w_c \, \overline{\text{conf}}(\pi) + w_m \, \overline{N}(\pi), \tag{4}$$

where $|\pi|/L_{\max}$ rewards longer chains, $\overline{\text{conf}}(\pi)$ (the mean edge confidence) rewards chains whose every step is well-evidenced, and $\overline{N}(\pi)$ (the mean MCTS visit count) rewards chains many rollouts converge on, following HopRAG [38]. To recover parallel evolutions at hubs, the lineage operator re-runs from every branch point with covered edges masked, yielding the final Π_q.

3.3.2 Graph-Grounded Idea Evaluation

Given an idea description d, the evaluator scores five dimensions and combines them into an overall score $s^*(d, G) \in [1, 10]$.

Per-dimension scoring. Each evaluation dimension probes a different structural property of d's methods inside G, so its score can be computed directly from graph statistics rather than text-based LLM reasoning. The evaluator first resolves the methods named in d into nodes $M_d \subseteq \mathcal{V}_M$ via the lookup table $\mathcal{A}$, then scores each of five dimensions (Novelty N, Feasibility F, Significance S, Validity V, Clarity C) independently from the position of M_d in d's retrieved context $\mathcal{C}_d$:

$$s_k(d, G) \;=\; \mathrm{clip}_{[1,10]}\Big(b_k + \sum_j w_j^{(k)} \cdot \phi_j^{(k)}(M_d, \mathcal{C}_d)\Big), \quad k \in \{N, F, S, V, C\}. \tag{5}$$

For each dimension k, the score combines a baseline b_k with a sum of weighted graph statistics, where each $\phi_j^{(k)}$ is a deterministic statistic computed from M_d and $\mathcal{C}_d$, and $w_j^{(k)}$ is its fixed scalar weight. One of the summed terms $\phi_j^{(F)}$ is an *anti-stacking* statistic that down-weights ideas dominated by overstudied methods. The complete catalogue of $\phi_j^{(k)}$ is detailed in Appendix C.3.

Cross-dimensional aggregation. Certain dimension combinations carry qualitative meaning that a weighted average cannot express. The evaluator then aggregates the five raw scores into a final score by combining a linear weighted sum with a non-linear correction:

$$s^*(d, G) \;=\; \mathrm{clip}_{[1,10]}\big(\mathbf{w}^\top \mathbf{s} + \Omega_{\text{cross}}(\mathbf{s})\big), \quad \mathbf{s} = (s_N, s_F, s_S, s_V, s_C). \tag{6}$$

Here $\mathbf{w}$ is a fixed weight vector and Ω_{cross} implements four hand-designed conjunctive penalties. A representative penalty fires when an idea scores high on Novelty but low on Feasibility, reflecting that novel but infeasible typically signals a flawed core proposal regardless of its Significance and Validity scores. Finally, an optional LLM reviewer may *only lower* s^*, providing a one-sided veto that catches occasional errors in the graph-based scoring without reintroducing the LLM's upward bias.

3.3.3 Strategy-Driven Idea Generation

While evaluation asks "how good is this idea?", generation asks the inverse question: "which new idea should be proposed?" Operating on the retrieval context $\mathcal{C}_q$ (q is a research method of interest), the generator scans for under-explored structural regions and returns a candidate research proposal d' tied to a concrete bottleneck the literature has yet to solve.

Structural gap patterns. From $\mathcal{C}_q$ and lineage chains Π_q, the generator extracts four structural patterns that together pinpoint where the literature has unfilled research opportunities. These patterns (*open axes*, *recent improvement directions*, *sacrifice axes*, and *disconnected pairs*) are bundled into a structural summary $\Phi(q, G)$ via deterministic graph queries (no LLM call).

Strategy-conditioned proposal. To translate Φ into a concrete proposal, we pair each pattern type with a dedicated generation strategy: *Bottleneck Resolution* for open axes, *Trend Extrapolation* for recent improvement directions, *Cross-Pollination* for disconnected pairs, and *Paradigm Challenge* for sacrifice axes. The chosen strategy σ fixes both the structural target and the proposal's high-level form, leaving the LLM only to fill in technical specifics:

$$d' \;=\; \mathrm{Generate}(\Phi(q, G), \sigma). \tag{7}$$

The strategy alone, however, does not stop the LLM from inventing a bottleneck and dressing it up in the right form. We close this loophole by requiring every proposal to carry an evidence certificate: a triple (e, b_e, j) where e is a specific causal edge from $\mathcal{C}_q$, b_e is the bottleneck text already stored in $\rho(e)$ (reproduced exactly), and j explains why this bottleneck is still unresolved. Before a proposal is returned, we exact-match b_e against the stored quote in $\rho(e)$; if it fails, the LLM output is discarded and a deterministic fallback emits a minimal valid proposal from graph content.

4 Experiments

Figure 3 provides a high-level view of the Intern-Atlas method landscape, showing major paradigm clusters and their evolution links. We evaluate Intern-Atlas from two perspectives: whether the graph reliably represents methodological evolution, and whether this structure is useful for downstream research tasks. For graph quality, Sec. 4.1 compares the constructed graph with survey-derived expert

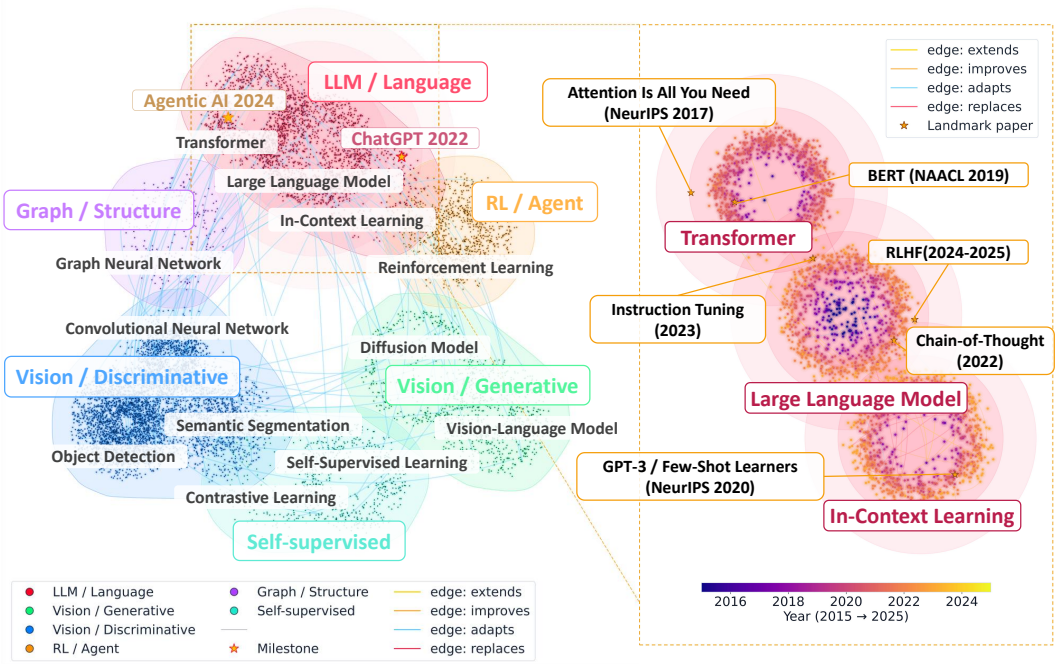


Figure 3: **Intern-Atlas method landscape.** *Left:* AI papers cluster into six paradigm continents connected by typed evolution edges (extends/improves/adapts/replaces). For visual clarity, only a representative subset of nodes is displayed. *Right:* Detailed view of the LLM continent with landmarks such as Attention, BERT, GPT-3, CoT, Instruction Tuning, and RLHF.

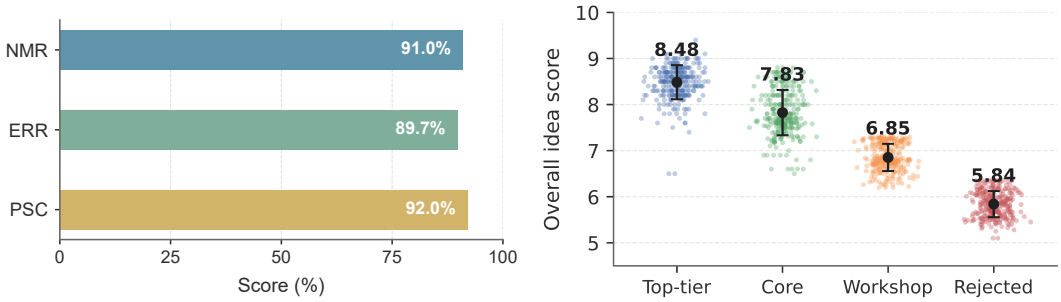
Table 1: Lineage-search algorithm performance. @k denotes beam width or RW rollouts; MCTS uses the same rollout budget as SGT-MCTS.

Method	NR	ER	CAS
Beam@1	41.0	18.6	41.0
Beam@5	43.4	21.6	43.4
Beam@10	44.9	23.2	44.9
RW@5	28.1	0.7	28.1
Vanilla MCTS	37.9	13.5	37.9
SGT-MCTS	**84.8**	**79.0**	**84.8**

references and tests whether SGT-MCTS can recover coherent method lineages from the graph. For downstream utility, Sec. 4.2 studies whether graph-derived evidence supports idea evaluation, while Sec. 4.3 studies whether retrieved method lineages help generate better research ideas. Together, these experiments evaluate Intern-Atlas at both the representation and application level.

4.1 Evaluating Graph Construction and Lineage Reconstruction

Method-evolution Benchmark and Metrics. We build a benchmark from 30 high-impact survey papers covering major AI subfields, yielding 30 method-evolution graphs (2,268 nodes, 1,462 edges, 133 chains). The 30 surveys span computer vision, NLP, RL, graph learning, and multimodal learning, averaging $\sim$76 nodes and $\sim$49 edges per survey, with chains reproduced from each survey author's own narrative. Graph construction is scored by Node Match Ratio (NMR; methods recovered as nodes), Edge Reachable Ratio (ERR; relations recovered as directed paths), and Path Semantic Correctness (PSC; correctness of recovered semantics). Lineage reconstruction is scored by Node Recall (NR), Edge Recall (ER), and Chain Alignment Score (CAS), covering reference methods, transitions, and ordering, respectively. Details are provided in Appendix D.1.



(a) Graph quality scores

(b) Overall tier scatter

Figure 4: **Graph construction quality and graph-grounded idea evaluation.** (a) Static graph quality on the method-evolution benchmark: node match ratio (NMR), edge reachable ratio (ERR), and path semantic correctness (PSC). (b) Distribution of Overall scores across strata. The dataset contains four groups (300 papers each). Black markers indicate the sample mean $\pm$ 1 s.d.

Table 2: Mean idea evaluation scores across publication strata. Scores are computed from extracted ideas. The overall score and all five dimensions decrease from top-tier conference papers to rejected submissions, with the largest gaps appearing in Significance and Validity.

	Novelty	**Feasibility**	**Significance**	**Validity**	**Clarity**	**Overall**
Top-tier conferences	**7.27**	**7.09**	**9.49**	**8.62**	**6.57**	**8.48**
Core conferences	6.54	6.80	7.16	6.47	6.52	7.83
Workshop papers	5.01	6.38	5.77	6.16	6.17	6.85
Rejected submissions	4.56	5.50	4.91	5.27	5.54	5.84

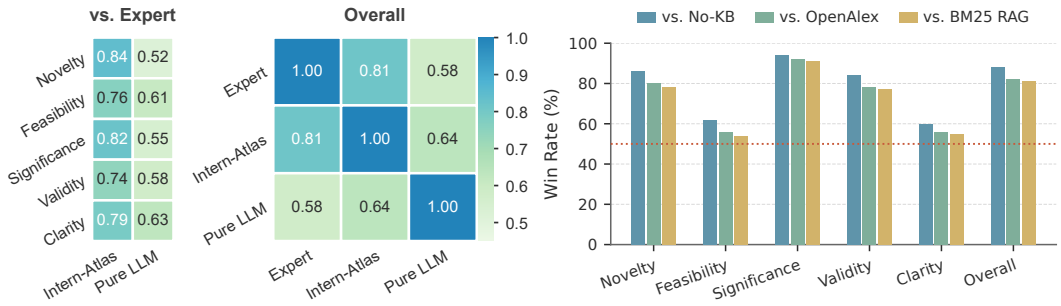
Evaluation of the SGT-MCTS algorithm. Table 1 compares SGT-MCTS with three baselines: Beam keeps the top-k locally scored partial chains, RW uniformly samples next hops, and vanilla MCTS uses standard UCT without the self-guided edge prior or guided rollout. Compared with the strongest baseline, Beam@10, SGT-MCTS improves NR by 39.9 points, ER by 55.8 points, and CAS by 39.9 points, indicating better recovery of the reference methods, transitions, and ordering in target evolution chains. These results support the effectiveness of SGT-MCTS for method-lineage reconstruction. Given the same graph and seed, SGT-MCTS finds chains closer to the method's expected evolutionary lineage than baselines.

Analysis of graph construction quality. Figure 4a reports graph quality on the benchmark: NMR 91.0%, ERR 89.7%, and PSC 92.0%. The high NMR and ERR show that Intern-Atlas captures most reference methods as nodes and most reference relations as recoverable directed paths, and PSC confirms that these paths carry the correct evolution semantics. Remaining mismatches stem from the survey-derived graphs and Intern-Atlas using different construction procedures and granularity. Section 4.4 compares the semantic correctness of paths obtained with three ways of constructing the links.

4.2 Evaluating the Graph-Grounded Idea Evaluator

Experimental details. We use the **Strata Dataset**: 1,200 papers (300 per stratum) drawn from top-tier conferences (ICLR 2026, ICML 2025, NeurIPS 2025), core conferences (AAAI 2026, IJCAI 2025), ICLR 2026 workshops, and ICLR 2026 rejections. For each paper we extract a concise idea summary and feed it to our evaluator (Sec. 3.3.2), which outputs five-dimensional plus overall scores. For each case, we use graph records available before the source paper's first public date and exclude the source paper itself (Appendix D.3).

Validation across publication strata. As shown in Table 2 and Figure 4b, the overall score follows the expected order: top-tier conference papers receive the highest mean score (8.48), followed by core conference papers (7.83), workshop papers (6.85), and rejected submissions (5.84). The largest gaps appear in *Significance* and *Validity*, where method-evolution structure most directly evidences



(a) Human-alignment correlations

(b) Pairwise generation win rate of Intern-Atlas

Figure 5: **Downstream utility of Intern-Atlas for idea evaluation and idea generation.** (a) Spearman correlations for idea evaluation: left, dimension-wise alignment of Intern-Atlas and a pure LLM judge with expert scores; right, overall-score correlations among expert judgments, Intern-Atlas, and the pure LLM baseline. (b) Pairwise win rates of Intern-Atlas against idea-generation baselines across five dimensions and overall quality.

whether an idea is tied to important, well-supported directions. *Novelty* also decreases across strata, while *Clarity* varies more mildly, since an extracted idea can read regardless of contribution depth.

Human evaluation. We sample 100 idea profiles and ask 10 AI PhD researchers to score them under the same five-dimensional rubric. The baseline is a pure LLM-as-judge given only the idea profile and rubric. Figure 5a shows that Intern-Atlas correlates with expert ratings more strongly than the LLM baseline on every dimension: 0.81 vs. 0.58 overall, with the largest gaps on *Novelty* (0.84 vs. 0.52) and *Significance* (0.82 vs. 0.55), where judging an idea requires comparison with prior methods and an understanding of its position in methodological development.

4.3 Evaluating Graph-Grounded Idea Generator

Experimental details. We use the pipeline of Sec. 3.3.3, varying only the background-knowledge source: **No-KB** (no grounding), **External Search** via OpenAlex / Semantic Scholar, and **Local RAG** on the same paper library as Intern-Atlas via BM25. Inputs are 100 research questions curated by 10 AI PhD researchers, who also provide blind pairwise win-rate judgments alongside our evaluator's scores. All five methods share the same LLM backbone and prompt template, so quality differences trace to the knowledge source rather than generation capacity. Human and automatic evaluators receive only the generated Idea fields, with evidence certificates and citation links removed. Details are provided in Appendix D.4.

Evaluation Results. Table 3 shows that Intern-Atlas achieves the best *Overall* score (7.20, $+1.02$ over the strongest baseline) and leads in *Novelty* (6.37), *Significance* (6.30), and *Validity* (6.26), with the largest gain on *Validity*, where method-evolution chains help the generator identify recurring limitations in prior work. *Feasibility* and *Clarity* are close across knowledge-grounded methods, which share the same LLM generator. Human blind pairwise judgments (Figure 5b) show the same trend: Intern-Atlas wins on overall quality by 88.0%, 82.0%, and 81.0% against No-KB, OpenAlex, and BM25 RAG. An additional comparison with Chain of Ideas (CoI) [33] on the same 100 queries yields 65.0% expert preference for Intern-Atlas. These results suggest that method-evolution context is more useful than document-level retrieval for generating high-quality research ideas. Three end-to-end case studies in Appendix E.2 qualitatively illustrate how the pipeline transfers across distinct research regimes.

Table 3: Idea-quality scores from our evaluator for generated ideas under different knowledge sources.

Method	Novelty	Feasibility	Significance	Validity	Clarity	Overall
No-KB	4.85	6.52	5.70	4.00	7.21	5.78
OpenAlex	5.27	7.08	5.34	4.56	7.52	6.03
Semantic Scholar	5.40	**7.32**	5.39	4.70	**7.75**	6.18
BM25 RAG	5.39	7.30	5.39	4.66	7.70	6.15
Intern-Atlas	**6.37**	7.21	**6.30**	**6.26**	7.67	**7.20**

4.4 Ablation Studies

Relation-layer ablation. We keep the matched method nodes fixed on the 30-survey benchmark and vary how their links are constructed. Path semantic correctness (PSC) measures the fraction of recovered paths that express the evolution relation described in the reference survey (Appendix D.1).

The raw-citation variant uses untyped citation links. The single-pass LLM assigns a relation type from the citation context. The full pipeline adds a second pass to extract the bottleneck and mechanism and checks the quoted evidence against the source text. In Table 4, relation typing raises PSC by 40.5 percentage points over raw citations. Completing and checking the evidence adds a further 8.8 points.

Table 4: Path semantic correctness on the survey benchmark.

Variant	PSC (%)
Raw citations	42.7
Single-pass LLM	83.2
Intern-Atlas	**92.0**

Additional analyses. The mean Overall score gap between top-tier and rejected papers is 2.64 points for the graph-based evaluator and 0.73 for the mean-paper-count baseline (Appendix D.3). Appendix C.4 examines how this gap changes when profiles with no matched methods are excluded.

5 Conclusion

We present INTERN-ATLAS, a graph of methodological evolution built from $1{,}030{,}314$ AI papers. Its $9{,}410{,}201$ typed relations support lineage reconstruction, idea evaluation, and idea generation. For causal relations, source passages document the bottleneck in an earlier method and how later work addressed it. SGT-MCTS recovers method order in survey-derived lineages more accurately than the search baselines. The evaluator's mean scores decrease from top-tier to rejected papers, and its ratings of individual idea profiles correlate with expert judgments. In blind comparisons, experts preferred ideas generated with Intern-Atlas to those generated with external search or local retrieval. A research agent can trace a proposed direction back to the methods and source passages that motivated it.

Acknowledgements and Disclosures

This work was supported by 2025 Local Supporting Project (PJ-PRJ24DATA001) by Shanghai Artificial Intelligence Laboratory. The authors declare no competing interests.

References

[1] Google scholar. `https://scholar.google.com`.

[2] Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, et al. Construction of the literature graph in semantic scholar. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 84–91, 2018.

[3] Jason Priem, Heather Piwowar, and Richard Orr. Openalex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. *arXiv preprint arXiv:2205.01833*, 2022.

[4] Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

[5] Aishik Sanyal, Samuel Schapiro, Sumuk Shashidhar, Royce Moon, Lav R Varshney, and Dilek Hakkani-Tur. Spark: A system for scientifically creative idea generation. *arXiv preprint arXiv:2504.20090*, 2025.

[6] Tianyu Fan, Fengji Zhang, Yuxiang Zheng, Bei Chen, Xinyao Niu, Chengen Huang, Junyang Lin, and Chao Huang. Deepinnovator: Triggering the innovative capabilities of llms. *arXiv preprint arXiv:2602.18920*, 2026.

[7] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[8] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[12] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[13] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[14] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[16] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022.

[17] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110, 2022.

[18] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[19] Norman P Hummon and Patrick Dereian. Connectivity in a citation network: The development of dna theory. *Social networks*, 11(1):39–63, 1989.

[20] Chaomei Chen. Citespace ii: Detecting and visualizing emerging trends and transient patterns in scientific literature. *Journal of the American Society for information Science and Technology*, 57(3):359–377, 2006.

[21] Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S Weld. S2orc: The semantic scholar open research corpus. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4969–4983, 2020.

[22] Xin Wang, Jiyao Liu, Yulong Xiao, Junzhi Ning, Lihao Liu, Junjun He, Botian Shi, and Kaicheng Yu. The-tree: Can tracing historical evolution enhance scientific verification and reasoning? *arXiv preprint arXiv:2506.21763*, 2025.

[23] Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers. *arXiv preprint arXiv:2409.04109*, 2024.

[24] Alina Beygelzimer, Yann N Dauphin, Percy Liang, and Jennifer Wortman Vaughan. Has the machine learning review process become more arbitrary as the field has grown? the neurips 2021 consistency experiment. *arXiv preprint arXiv:2306.03262*, 2023.

[25] Yansheng Qiu, Haoquan Zhang, Zhaopan Xu, Ming Li, Diping Song, Zheng Wang, and Kaipeng Zhang. Ai idea bench 2025: Ai research idea generation benchmark. *arXiv preprint arXiv:2504.14191*, 2025.

[26] Sikun Guo, Amir Hassan Shariatmadari, Guangzhi Xiong, Albert Huang, Myles Kim, Corey M Williams, Stefan Bekiranov, and Aidong Zhang. Ideabench: Benchmarking large language models for research idea generation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 5888–5899, 2025.

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

[28] Xuemei Gu and Mario Krenn. Interesting scientific idea generation using knowledge graphs and llms: Evaluations with 100 research group leaders. *arXiv:2405.17044*, 2024.

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

[30] Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang, and Linyi Yang. Cycleresearcher: Improving automated research via automated review. *arXiv preprint arXiv:2411.00816*, 2024.

[31] Jiakang Yuan, Xiangchao Yan, Botian Shi, Tao Chen, Wanli Ouyang, Bo Zhang, Lei Bai, Yu Qiao, and Bowen Zhou. Dolphin: Closed-loop open-ended auto-research through thinking, practice, and feedback. *arXiv e-prints*, pages arXiv–2501, 2025.

[32] Zijun Liu, Kaiming Liu, Yiqi Zhu, Xuanyu Lei, Zonghan Yang, Zhenhe Zhang, Peng Li, and Yang Liu. Aigs: Generating science from ai-powered automated falsification. *arXiv preprint arXiv:2411.11910*, 2024.

[33] Long Li, Weiwen Xu, Jiayan Guo, Ruochen Zhao, Xingxuan Li, Yuqian Yuan, Boqiang Zhang, Yuming Jiang, Yifei Xin, Ronghao Dang, et al. Chain of ideas: Revolutionizing research via novel idea development with llm agents. *arXiv preprint arXiv:2410.13185*, 2024.

[34] Xiang Hu, Hongyu Fu, Jinge Wang, Yifeng Wang, Zhikun Li, Renjun Xu, Yu Lu, Yaochu Jin, Lili Pan, and Zhenzhong Lan. Nova: An iterative planning and search approach to enhance novelty and diversity of llm generated ideas. *arXiv preprint arXiv:2410.14255*, 2024.

[35] Qingyun Wang, Doug Downey, Heng Ji, and Tom Hope. Scimon: Scientific inspiration machines optimized for novelty. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 279–299, 2024.

[36] Stephen Robertson and Hugo Zaragoza. *The probabilistic relevance framework: BM25 and beyond*, volume 4. Now Publishers Inc, 2009.

[37] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer, 2006.

[38] Hao Liu, Zhengren Wang, Xi Chen, Zhiyu Li, Feiyu Xiong, Qinhan Yu, and Wentao Zhang. Hoprag: Multi-hop reasoning for logic-aware retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1897–1913, 2025.

Appendix Table of Contents

A Graph Construction **15**

A.1 Schema 15

A.2 Corpus, PDF Pipeline, and Reference Resolution 16

A.3 Two-Phase Extraction Protocol 17

B Lineage Reconstruction (SGT-MCTS) **17**

C Idea Evaluation **18**

C.1 Common Retrieval Routine 18

C.2 Duplicate-Risk Detection 18

C.3 Per-Dimension Signal Specifications 18

C.4 Graceful Fallback and Optional Adjudication 18

D Additional Experimental Details **19**

D.1 Method-Evaluation Benchmark and Metrics 19

D.2 Lineage-Search Baselines 20

D.3 Dataset and Baseline Details for Evaluating the Idea Evaluator 20

D.4 Experimental Details for Evaluating Idea Generation 21

E Case Studies **22**

E.1 Case Study on Lineage Search 22

E.2 End-to-End Case Studies on Idea Generation 23

F Limitations and Broader Impact **25**

A Graph Construction

A.1 Schema

Node types. $\mathcal{V}$ is partitioned into three disjoint classes:

- Paper nodes $\mathcal{V}_P$: parsed full texts within the collection window, with resolved references.
- Method entity nodes $\mathcal{V}_M$: canonical methods, each tagged with a canonical name; surface-form mapping is handled by the alias registry $\mathcal{A}$ below.
- Stub nodes $\mathcal{V}_S$: cited papers outside the collection window, retained as metadata-only placeholders so that historical lineages remain reachable during SGT-MCTS.

The publication-year map $\tau : \mathcal{V}_P \cup \mathcal{V}_S \to \mathbb{Z}$ is populated from DOI/OpenAlex metadata; missing entries are treated as a neutral default by the temporal-coherence function in Appendix B.

Bottleneck dimension taxonomy $\mathcal{D}$. The 14 axes in Table 5 were bootstrapped by clustering 500 randomly sampled bottleneck quotes from a pilot run on NeurIPS-2024 full texts and consolidated by the authors into a minimal spanning set. Their operational definitions are reproduced verbatim in the Phase-2 extraction prompt and are used wherever a record field is typed by $\mathcal{D}$.

Table 5: The 14-axis bottleneck dimension taxonomy $\mathcal{D}$. Each axis names a fundamental cost or quality of methodological design; axes type the bottleneck and impact fields of $\rho(e)$ and feed the feasibility/significance signals of the idea evaluator. Definitions are reproduced verbatim in the Phase-2 extraction prompt.

Dimension	Operational definition
computational complexity	asymptotic or wall-clock compute at fixed scale
memory efficiency	peak activation / parameter memory footprint
parallelization	degree of across-device or across-token parallelism
accuracy	task-level correctness or quality metric
generalization	out-of-distribution / cross-domain transfer
scalability	behavior as model / data / context size grows
data efficiency	sample complexity at fixed quality target
training stability	variance / divergence risk during optimization
inference speed	serving-time latency or throughput
expressiveness	function class or representational capacity
simplicity	implementation, conceptual, or interface simplicity
robustness	behavior under perturbation or adversarial input
hyperparameter sensitivity	outcome variance w.r.t. hyperparameter choice
training complexity	engineering difficulty of the training recipe

Edge vocabulary $\mathcal{T}$. Edges $\mathcal{E} \subseteq (\mathcal{V}_P \cup \mathcal{V}_S)^2$ carry a type from a seven-class edge vocabulary (Table 6), whose glossary is reproduced verbatim in the Phase-1 extraction prompt.

Strong-causal subgraph G_{strong}. $\mathcal{T}_{\text{strong}} = \{\texttt{extends}, \texttt{improves}, \texttt{replaces}, \texttt{adapts}\}$ induces $G_{\text{strong}} = (\mathcal{V}_P \cup \mathcal{V}_S, \mathcal{E}_{\text{strong}}, \tau, \rho)$. Excluding `uses_component` separates methodological evolution (inheriting the parent mechanism) from compositional reuse (borrowing only a module). Component reuse is preserved inside the method-level DAG G_M for evaluation and generation but does not drive lineage traversal.

Method-level DAG G_M. $G_M = (\mathcal{V}_M, \mathcal{E}_M)$ is derived from $\mathcal{E}$ by the deterministic projection in Table 7. Two further relations, `optimizes` and `inspired_by`, are not produced by citation-anchored extraction: `optimizes` typically surfaces in release-note or blog text lacking a formal citation, while `inspired_by` appears in narrative framing rather than bibliographic anchors. We source both from hand-curated seed annotations, yielding $\mathcal{T}_M = \{\texttt{variant_of}, \texttt{specializes}, \texttt{component_of}, \texttt{optimizes}, \texttt{inspired_by}\}$.

Table 6: The seven edge types of $\mathcal{T}$. Definitions are reproduced verbatim in the Phase-1 extraction prompt; the first four form the strong-causal subset $\mathcal{T}_{\text{strong}}$ on which SGT-MCTS performs lineage traversal.

Edge type	Operational definition (Phase-1 prompt glossary)
`extends`	adds a new capability or component on top of the cited method
`improves`	optimizes an existing component along ≥ 1 dimension in $\mathcal{D}$ without altering the core formulation
`replaces`	substitutes a load-bearing component with a qualitatively different mechanism
`adapts`	ports the cited method to a new domain, modality, or task setting
`uses_component`	re-uses the cited method as an auxiliary module, not as the central contribution
`compares`	cites purely as a baseline or head-to-head comparison
`background`	cites as context, motivation, or non-methodological background

Table 7: Deterministic projection from paper-level edges types $\mathcal{T}$ to method-level relations $\mathcal{T}_M$. `compares` and `background` are excluded as non-methodological, and `uses_component` is reversed in direction so that the cited method is encoded as a component of the citing one.

Edge type in $\mathcal{T}$	Method-level type in $\mathcal{T}_M$
`extends`, `improves`	`variant_of`
`adapts`, `replaces`	`specializes`
`uses_component`	`component_of` (direction reversed)
`compares`, `background`	excluded (non-methodological)

Alias registry $\mathcal{A}$. $\mathcal{A} : \Sigma^* \to \mathcal{V}_M$ maps each normalized surface form to a canonical method node. The lookup procedure is substring-based with case/punctuation normalization, word-boundary enforcement (preventing "GPT" matching inside "lgpto"), longest-match priority ("GPT-4 Turbo" over "GPT-4" over "GPT"), version-suffix consolidation ("-v2", "-Large", etc. attach to the parent unless a distinct canonical node exists), and a manually curated negative-surface list for ambiguous names (e.g., "Mamba" the state-space model versus the Python linter). The registry contains $8{,}155$ canonical methods and $9{,}545$ aliases. Construction details are in §A.2.

A.2 Corpus, PDF Pipeline, and Reference Resolution

Collection window and venues. Intern-Atlas indexes a comprehensive corpus of major AI venues, journals, and preprint servers over the window 1965–early 2026, spanning core ML (NeurIPS, ICML, ICLR), computer vision (CVPR, ICCV, ECCV), NLP (ACL, EMNLP, NAACL), general AI (AAAI, IJCAI), and data/information (KDD, SIGIR). ICCV and ECCV run in alternating years, explaining their narrower temporal footprint; all other venues contribute three editions each. Main-conference, Findings (for ACL/EMNLP/NAACL), and accepted-workshop tracks are included uniformly.

PDF parsing pipeline. Full texts are obtained from the official proceedings (where openly licensed) and from OpenReview / arXiv preprints as fallbacks. Each PDF is parsed by NOUGAT-V1.0 for born-digital PDFs, with GROBID-V0.8.0 as a fallback for scanned or degraded files that Nougat fails to parse (3.7% of the corpus). The parser emits structured JSON with section headings, paragraph-level text, and a reference list. Downstream extraction consumes only the `Introduction`, `Method/Approach`, and `Related Work` sections of the citing paper; for cited references only the title and abstract are consumed in Phase 1, and the title plus parsed bibliographic metadata in Phase 2.

Method entity curation. $\mathcal{V}_M$ is populated in two stages. First, a seed list of 247 widely-used methods is hand-curated by combining the Papers With Code method ontology with mentions enumerated in 18 pilot review articles spanning 2020–2024. Second, an LLM expansion pass scans full-text Method sections for proper-noun method names following capitalization and acronym heuristics; candidates are retained iff they co-occur with ≥ 3 distinct papers' Method sections and pass adjudication by one of the authors.

A.3 Two-Phase Extraction Protocol

Extraction decomposes into a high-recall Phase-1 classification over $\mathcal{T}$ followed by a Phase-2 record-completion pass over non-background candidates. We use QWEN3.6-35B-A3B as the production extraction model for both phases and CLAUDE-SONNET-4.6 as the audit model used only in the reliability benchmarks. All calls use temperature 0 and deterministic decoding.

Phase 1 (classification). The model receives the citing paper's Introduction and Method sections, a list of references with title and abstract, and the verbatim glossary of Table 6. For each reference, it returns a JSON record `{type, evidence_snippet}`, with `background` reserved for citations that describe no methodological relation. Phase 1 is tuned for high recall: the prompt explicitly instructs the model to prefer a non-background type over `background` in borderline cases, and the downstream validator, together with Phase 2, suppresses false positives. Each prompt batches all references from one citing paper (mean 21.2, max 96); mean input/output lengths are 5.8k / 2.1k tokens.

Phase 2 (record completion). Phase 2 consumes only the non-background candidates from Phase 1, batched 10 at a time together with the citing paper's Method section and the verbatim taxonomy of Table 5. For each (citing, cited) pair already typed, the model fills in the structured record $\rho(e)$ ={`bottleneck:{description, severity, verbatim_quote, dimension}, mechanism:{description, type, verbatim_quote}, impact:{improvement_dim, sacrifice_dim, tradeoff_sentence}, confidence}`, with all `verbatim_quote` fields required to be copied verbatim from the citing paper's text. Malformed or schema-violating batches are retried once with a repair prompt; a second failure drops the batch and logs it for manual inspection. Mean input/output lengths are 7.9k / 2.8k tokens.

B Lineage Reconstruction (SGT-MCTS)

Hyperparameters. $c = \sqrt{2}$ (UCT exploration), $\lambda = 0.5$ (prior weight), per-direction-per-seed budget $B = 200$, rollout depth cap $d_{\max} = 5$, top-K primary lineages $K = 5$, deduplication node-Jaccard threshold $\theta_{\text{dup}} = 0.8$, dead-end backpropagation penalty $\eta = -0.05$.

Temporal coherence.

$$\mathrm{TC}(\Delta\tau) = \begin{cases} 0.40 & -1 \le \Delta\tau < 0 \\ 0.85 & \Delta\tau = 0 \\ 1.00 & 1 \le \Delta\tau \le 3 \\ 0.80 & 4 \le \Delta\tau \le 6 \\ \max(0.30,\ 1.00 - 0.08(\Delta\tau - 6)) & \Delta\tau > 6 \\ 0.70 & \tau \text{ missing} \end{cases} \tag{8}$$

Edges with $\Delta\tau < -1$ in the search direction are hard-filtered prior to evaluation.

Algorithm. For each seed $s \in \mathcal{S}(q)$ and each direction in $\{\text{back}, \text{fwd}\}$, MCTS runs for B iterations on G_{strong}. Each iteration follows four standard phases: (i) selection by SGT-UCT (Eq. 3); (ii) expansion of the highest-confidence untried child after discarding cycles and edges violating the hard temporal filter; (iii) greedy rollout that at each step picks the child maximizing $\mathrm{conf} \cdot \mathrm{TC}$ and terminates at $d_{\max}$, a leaf, or a cycle, scored by $R(\pi)$; (iv) backpropagation along the selection path, with an additional penalty η to ancestors of dead-end leaves. The two trees per seed are spliced through s by concatenating each of the top-5 backward paths (ranked by accumulated Q) with each of the top-5 forward paths; the resulting candidates are deduplicated under θ_{dup}.

Branch discovery. After the top-K primary lineages are finalized, SGT-MCTS is re-run from each *branch point*, defined as a node with ≥ 2 strong-causal children in the search direction, only one of which was traversed by the primary set. Each re-run masks already-covered edges, uses budget $B/2$, and contributes its surviving lineages to the output pool under the same deduplication rule, surfacing parallel evolutionary trajectories that pure greedy traversal would collapse.

C Idea Evaluation

C.1 Common Retrieval Routine

Context retrieval. Given x (an idea d or a query q), the routine returns the tuple (M_x, P_x, E_x, B_x):

1. $M_x \subseteq \mathcal{V}_M$: canonical methods resolved from x via the alias registry $\mathcal{A}$.
2. $P_x \subseteq \mathcal{V}_P$: the top-$K$ papers ($K = 500$) under a hybrid score combining alias-count over M_x with BM25 keyword relevance on the remaining content words.
3. $E_x \subseteq \mathcal{E}$: non-`background` causal edges incident on P_x.
4. B_x: bottleneck records aggregated from $\rho(E_x)$.

C.2 Duplicate-Risk Detection

A separate retrieval stack, invoked only by the Novelty signal, ranks $\mathcal{V}_P$ against d in three stages. Stage 1 (*candidate pooling*) fuses a dense ranking from BGE-SMALL-EN-V1.5 (384-dim cosine similarity) with a sparse BM25-Okapi ranking ($k_1 = 1.5$, $b = 0.75$) via Reciprocal Rank Fusion ($k_{\text{RRF}} = 60$), yielding the top-20 candidates. Stage 2 (*rerank*) re-scores those candidates with the MS-MARCO-MINILM-L-12-V2 cross-encoder on (d, p) pairs. Stage 3 (*final score*) returns $\text{fused}(d,p) = 0.5\,\text{dense}(d,p) + 0.5\,\sigma\big(\text{rerank_logit}(d,p)\big)$; the sparse signal contributes only through Stage 1 pooling and does not enter the final fusion. Writing $s = \max_p \text{fused}(d,p)$, the Novelty signal applies a step penalty of -0.5 at $s \geq 0.55$, -1.5 at $s \geq 0.65$, -2.5 at $s \geq 0.75$, and -4.0 at $s \geq 0.85$ (zero otherwise).

C.3 Per-Dimension Signal Specifications

Bases and weights. All five dimensions share base $b_k = 5.0$. Aggregation weights are $\mathbf{w} = (w_N, w_F, w_S, w_V, w_C) = (0.20, 0.20, 0.25, 0.20, 0.15)$. Signal contributions are additive on b_k; the sum is clipped to $[1, 10]$.

Per-dimension signals. Load-bearing components are listed below.

Novelty. (i) Topological disconnection of M_d in the method-co-utilization graph, up to $+2.0$; (ii) mechanism-level Jaccard distance against E_d, $+[0, 1.5]$; (iii) frontier-leaf bonus, $+0.8$; (iv) duplicate-risk penalty $-\text{pen}(d) \in [-4.0, 0]$ from §C.1.

Feasibility. A sweet-spot maturity curve over each method's paper-count $\text{pc}(m)$,

$$\phi_F(m) = \begin{cases} 1.5 + 1.5 \cdot \frac{\text{pc}(m)}{500} & \text{pc}(m) \leq 500 \\ 3.0 - 1.0 \cdot \frac{\text{pc}(m) - 500}{1500} & 500 < \text{pc}(m) \leq 2000 \\ 1.5 & \text{pc}(m) > 2000 \end{cases}$$

averaged over M_d and capped at $+3.5$. The $500/2000$ thresholds are the 60th and 90th percentiles of $\text{pc}(\cdot)$ on the corpus; the non-monotonic shape blocks mashups from inflating F via pervasive terms. Auxiliary signals: full-text resource availability and a complexity penalty for $|M_d| \geq 4$.

Significance. Time-decayed in-degree on P_d (5-year half-life, non-background); frontier presence (≥ 3 non-background out-edges since 2021); and the *method-frontier regularizer*, the dominant anti-mashup term, with mean popularity over $M_d[:5]$ contributing $+2.5$ when < 300 (niche) and decaying linearly to -2.0 at > 1000 (saturated).

Validity. Bottleneck grounding against B_d (matched dimension tag plus a description bigram, up to $+3.5$); ancestry consistency in G_M within depth 4; type-weighted edge density on E_d.

Clarity. Method recognition rate; specificity peak at $|M_d| \in \{2, 3\}$ decaying for $|M_d| \geq 6$; structural completeness across {problem, approach, target}; length adequacy on 20–200 words.

C.4 Graceful Fallback and Optional Adjudication

Graceful fallback. When no method in an idea profile matches the registry ($M_d = \emptyset$), the evaluator returns the population prior $s^* = 6.5$. In a pilot of 147 ideas from NeurIPS 2024 submissions,

Table 8: Four empirical cross-dimensional priors comprising Ω_{cross}. Each row triggers an additive adjustment to the overall score s^* when its conjunctive condition over the post-red-flag scores $\mathbf{s}'$ holds, encoding patterns where one dimension's score becomes more or less informative in conjunction with another.

Prior	Trigger	Adjustment
Ideation–execution gap	$s'_N \geq 7 \ \wedge \ s'_F < 4$	-0.6
Validity–feasibility coherence	$s'_V \geq 7 \ \wedge \ s'_F \geq 7$	$+0.2$
Significance amplification	$s'_S \geq 6$	$+0.4$
	$5 \leq s'_S < 6$	$+0.2$
Balanced coherence	$\max(\mathbf{s}') - \min(\mathbf{s}') \leq 2 \ \wedge \ \min(\mathbf{s}') \geq 5$	$+0.3$

text-only signals for profiles with $M_d = \emptyset$ had a Spearman correlation of $\rho = 0.08$ ($p = 0.31$) with peer-review scores.

On the Strata Dataset, 36 of 1,200 profiles (3.0%) received the fallback score. Of the 300 profiles in each stratum, the counts were 6 for top-tier papers, 8 for core conferences, 10 for workshops, and 12 for rejected papers. Excluding these profiles gives mean Overall scores of 8.52 for top-tier papers and 5.81 for rejected papers. The gap widens from 2.64 to 2.71 points.

Adjudicator (optional). When enabled, two LLM calls modify s^* via bounds rather than recompute it. *Part A* (duplicate verification): the adjudicator classifies the relationship between d and the top-3 retrieved candidates as {duplicate, related, unrelated}; the strongest verdict determines the restoration rate of the automatic duplicate-risk penalty: 0% for duplicate (full penalty retained), 60% for related, 90% for unrelated. *Part B* (per-dimension upper bound): the adjudicator rates the idea on coherence, novelty, validity, and claim plausibility on 1–10; each rating acts as an upper bound on its mapped Atlas dimension via $s_k \leq \text{llm_score}_k + 1.0$, with coherence $\rightarrow \{V, C\}$, novelty $\rightarrow N$, plausibility $\rightarrow \{F, S\}$. Any sub-score below 3 triggers a hard cap $s^* \leq 6.0$. The adjudicator therefore never computes s^* *de novo*, preserving the zero-trainable-parameter property of the core scorer.

D Additional Experimental Details

D.1 Method-Evaluation Benchmark and Metrics

Purpose and selection. We construct the method-evolution benchmark from 30 high-impact survey papers, using surveys as proxies for expert consensus on methodological evolution. Candidate surveys are selected according to five criteria: authority, impact, classic status, subfield diversity, and method-centricity. In practice, we prioritize surveys published in recognized venues, widely cited by the community, commonly used as field-level taxonomies, covering diverse AI subfields, and containing explicit discussions of method evolution rather than only application summaries. These surveys provide external references for both static graph coverage and lineage reconstruction evaluation.

Construction. Each survey is processed in two stages. First, an LLM extracts discussed methods, directed evolution relations, and narration-based method chains from the survey text and taxonomy figures. Second, domain researchers manually audit the extracted results by removing hallucinated methods, correcting relation directions, merging duplicate method names, and checking each retained relation against the original survey text or figure. The final benchmark contains 30 survey-derived method-evolution graphs with 2,268 nodes, 1,462 directed evolution edges, and 133 reference evolution chains.

Evaluation protocols. We evaluate the benchmark in two parts, following the main text.

1. Static Graph Coverage. Given a survey-derived graph G^* and Intern-Atlas graph G, we report Node Match Ratio (NMR), Edge Reachable Ratio (ERR), and Path Semantic Correctness (PSC). NMR measures the fraction of survey methods matched to Intern-Atlas nodes. ERR measures the fraction of reference edges that can be recovered as directed paths in G. PSC measures whether reachable paths

preserve the intended method-evolution semantics, assessed by GPT-4.5 (LLM-as-judge) against the survey reference.

2. Lineage Reconstruction. For each reference chain, every search method starts from the same seed and returns a candidate chain. We report Node Recall (NR), Edge Recall (ER), and Chain Alignment Score (CAS). NR measures how many reference methods are recovered. ER measures how many adjacent reference transitions are recovered. CAS measures how well the retrieved chain preserves the ordering of the reference chain.

D.2 Lineage-Search Baselines

To evaluate the lineage reconstruction capability of SGT-MCTS, we compare it against three representative graph-search baselines. All methods operate on the same knowledge graph, start from the same query seeds, and use the same depth constraint.

- **Beam Search:** Expands valid successors and keeps the top-k locally scored partial paths at each step. We evaluate beam widths $k \in \{1, 5, 10\}$.
- **Random Walk (RW):** Samples valid outgoing edges uniformly until reaching the depth cap or a dead end. We report RW@5 with five independent rollouts per task. Additionally, we report RW@100 in the case study as a stronger implementation.
- **Vanilla MCTS:** Uses the standard UCT tree policy under the same rollout budget as SGT-MCTS, but without the self-guided edge prior or guided rollout policy.

All retrieved lineages are evaluated against the same reference chains and metric suite described previously.

D.3 Dataset and Baseline Details for Evaluating the Idea Evaluator

To validate the graph-grounded idea evaluator, we construct the Strata Dataset and apply a unified extraction protocol that standardizes the input format across all compared methods. We use the full dataset for publication-strata analysis and a 100-sample expert-rated subset for human-alignment evaluation.

The Strata Dataset. This dataset tests whether the evaluator's outputs follow the broad quality ordering reflected by different publication strata. We collected 1,200 papers evenly split across four tiers (300 each): (1) **Top-tier AI conferences**: accepted papers from ICLR 2026, ICML 2025, and NeurIPS 2025; (2) **Core AI conferences**: papers from AAAI 2026 and IJCAI 2025; (3) **Workshop papers**: submissions accepted to ICLR 2026 workshops; and (4) **Rejected submissions**: papers declined at ICLR 2026. All texts and metadata were obtained from OpenReview and publicly available proceedings. Venue and outcome labels are used only for aggregate analysis and are not provided to the evaluator.

Temporal cutoff. For each evaluated paper, we restrict scoring to paper and method nodes that were public before its first public date. We use the forum creation date for OpenReview papers and the first arXiv or proceedings release date for other papers. The evaluated paper itself and records that became public after its cutoff are excluded from scoring.

Human-Rated Subset. For human-alignment evaluation, we sample 100 idea profiles from the Strata Dataset and ask 10 AI PhD researchers to rate each profile on a 1-10 scale across the same five dimensions used by Intern-Atlas: novelty, feasibility, significance, validity, and clarity. The experts are active AI researchers and are shown only the extracted idea profiles, not the paper source, venue, acceptance outcome, Intern-Atlas scores, or LLM baseline scores. We average expert ratings for each profile and compute Spearman correlations between the expert scores and the scores produced by Intern-Atlas and the pure LLM-as-judge baseline.

Idea Extraction Protocol. Before scoring, we extract a standardized summary from each paper to ensure the evaluator focuses on the research idea itself rather than writing style. Specifically, given a paper's abstract, introduction, and core method sections, an LLM-based extractor produces a structured *Idea Profile* with four fields: (1) **Problem Formulation**: the specific limitation or

bottleneck the paper addresses; (2) **Core Innovation**: the key idea or mechanism proposed; (3) **Technical Implementation**: the main algorithmic components needed to realize the idea; and (4) **Target Setting**: the task or application domain. The extractor is prompted to use a neutral tone and omit background exposition, rhetorical framing, and unsubstantiated claims, retaining only the factual methodological content. The evaluator then operates on this extracted profile rather than the raw paper text, which keeps the input format consistent across all compared methods and avoids confounding the evaluation with writing quality.

Pure LLM-as-judge Baseline. This baseline employs Qwen3.5-397B-A17B to evaluate the standardized *Idea Profile* in a zero-shot setting. It is prompted with the exact same five-dimensional 1–10 scoring rubric used by Intern-Atlas and the human experts, but relies strictly on its internal parametric knowledge. It computes the evaluation without access to external document retrieval engines or any structured method-evolution graph context.

Mean-paper-count control. We use the same alias registry as the graph-based evaluator to identify the methods M_d in each idea profile. The baseline counts the papers associated with each of these methods, averages the counts within the profile, and normalizes the result to the 1–10 scale. Both sets of mean scores decrease across the four strata (Table 9). The difference between the top-tier and rejected groups is $7.62 - 6.89 = 0.73$ points for the count baseline and $8.48 - 5.84 = 2.64$ for Intern-Atlas.

Table 9: Mean Overall scores by publication stratum on the Strata Dataset, with 300 profiles per group.

Evaluator	Top-tier	Core	Workshop	Rejected
Mean-paper-count-only	7.62	7.41	7.35	6.89
Intern-Atlas	8.48	7.83	6.85	5.84

D.4 Experimental Details for Evaluating Idea Generation

Query Construction and Distribution. To standardize the evaluation of automated idea generation, we curated a diverse set of 100 high-quality research queries. To ensure clear constraints and evaluation objectives, each query follows a strict tripartite structure:

- **Domain Setting:** Establishes the specific AI subfield (e.g., *"In retrieval-augmented generation..."*).
- **Core Challenge:** Identifies the fundamental bottleneck or limitation of existing methods (e.g., *"...multi-step retrieval is fundamentally restricted by high resource cost..."*).
- **Expectation:** Specifies the desired capabilities of the target solution (e.g., *"Therefore, there is a critical need to build a resource-efficient mechanism..."*).

To comprehensively evaluate the generative generalizability across different topics, these 100 queries are distributed across six major AI fields: NLP & LLMs (34%), General ML & Optimization (19%), Computer Vision (14%), Reinforcement Learning & Agents (14%), Multimodal & Speech (13%), and Graph Learning (6%). The full list of queries is provided in the supplementary material.

Chain of Ideas baseline. We run Chain of Ideas (CoI) [33] on the same 100 research queries, following its original retrieval and paper-chain/idea-chain construction procedure. Its proposals are evaluated using the blind expert pairwise protocol below.

Human Expert Win-Rate Evaluation. To provide a rigorous, unbiased assessment of idea quality and compute the win rates, we assembled a panel of human experts, comprising active researchers with publication records in top-tier AI venues. The evaluation followed a strict double-blind pairwise comparison protocol. For each query, the experts were presented with the reference query alongside anonymized idea proposals generated by Intern-Atlas and the baselines. Evidence certificates are checked during generation. Before human or automatic evaluation, the certificates and specific citation links are removed, and evaluators receive only the generated Idea fields in a common format.

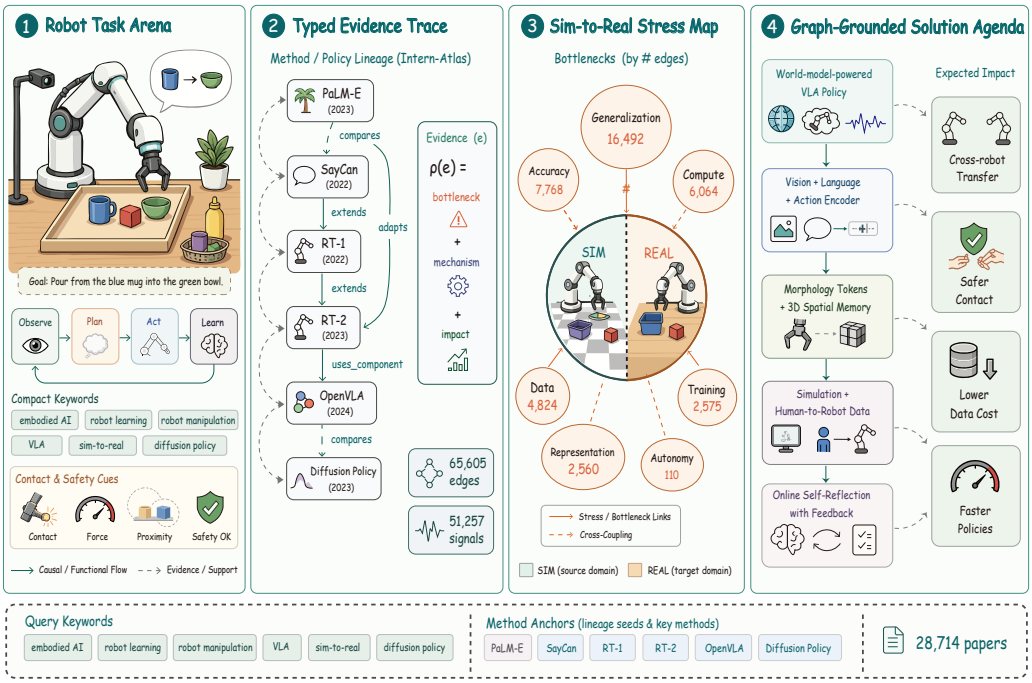


Figure 6: **Sim-to-real robot manipulation.** Given a tabletop goal (*Pour from the blue mug into the green bowl*) and six method anchors covering language-grounded planning, end-to-end VLA policies, and diffusion controllers, Intern-Atlas surfaces 28,714 candidate papers across four panels: **(1)** the task arena with compact query keywords and contact/safety cues; **(2)** a typed method-lineage trace PALM-E→SAYCAN→RT-1→RT-2→OPENVLA→DIFFUSION POLICY over 65,605 typed edges, each carrying an evidence record $\rho(e)$; **(3)** the sim-to-real stress map with bottlenecks dominated by *generalization* (16,492 signals) and *accuracy* (7,768); **(4)** a graph-grounded five-stage agenda culminating in a world-model-powered VLA proposal whose expected impact (cross-robot transfer, safer contact, lower data cost, faster policies) is pinned to specific bottleneck channels.

They were tasked with selecting the superior idea (or declaring a tie) based on overall methodological quality and five-dimensional rubric. Crucially, experts were permitted to consult external academic search tools during the evaluation to verify factual correctness and the true novelty of the proposed mechanisms.

E Case Studies

E.1 Case Study on Lineage Search

We use a familiar ConvNet lineage as an illustrative example of how different search algorithms behave on a high-branching architecture graph. All searches are seeded at *ConvNeXt V2*. The reference sketch starts from *AlexNet* as an early ImageNet ConvNet milestone and then follows the residual ConvNet backbone: *VGG* scales ConvNets through deeper stacks of small convolutions; *ResNet* addresses the optimization difficulty of very deep networks with residual mappings; *ResNeXt* extends residual blocks with aggregated transformations; *ConvNeXt* modernizes ResNet-style ConvNets using Transformer-era design choices; and *ConvNeXt V2* further extends ConvNeXt with a fully convolutional masked autoencoder and global response normalization.

Reference: *AlexNet* → *VGG* → *ResNet* → *ResNeXt* → *ConvNeXt* → *ConvNeXt V2*.

SGT-MCTS skips the very early *AlexNet* root but preserves the main backbone (*VGG* → *ResNet* → *ResNeXt* → *ConvNeXt* → *ConvNeXt V2*) in order. The extra *Caffe* node reflects implementation context before *VGG*, and the final node is a downstream user of *ConvNeXt V2*; neither changes the central lineage. Beam search recovers the later segment from *ResNet* to *ConvNeXt V2*, but

Table 10: Qualitative comparison of search results for a ConvNet lineage seeded at ConvNeXt V2.

Method	Retrieved lineage
Reference	*AlexNet* → *VGG* → *ResNet* → *ResNeXt* → *ConvNeXt* → *ConvNeXt V2*
SGT-MCTS	*Caffe* → *VGG* → *ResNet* → *ResNeXt* → *ConvNeXt* → *ConvNeXt V2* → *Robust Context-Aware Object Recognition*
Beam@10	*Speech Recognition with Deep RNNs* → *LSTM ASR* → *Highway Networks* → *ResNet* → *ResNeXt* → *ConvNeXt* → *ConvNeXt V2*
Vanilla MCTS	*VGG* → *ResNet* → *ConvNeXt* → *ConvNeXt V2*
RW@100	*Learning to Forget* → *Highway Networks* → *ResNet* → *Shake-Shake* → *Decoupled Weight Decay* → *ConvNeXt* → *ConvNeXt V2*

misses the early depth-scaling context and enters through an RNN/Highway-Network branch. Vanilla MCTS recovers *VGG*, *ResNet*, *ConvNeXt*, and *ConvNeXt V2*, but skips the *ResNeXt* aggregation step. Random walk recovers only a shorter local subsequence around *ResNet*, *ConvNeXt*, and *ConvNeXt V2*, while skipping *AlexNet*, *VGG*, and *ResNeXt*. This case illustrates the role of self-guidance: it keeps search concentrated on the ordered ConvNet backbone while reducing drift into side branches or less central intermediates.

E.2 End-to-End Case Studies on Idea Generation

To illustrate how Intern-Atlas operates as a unified infrastructure across heterogeneous research domains, we present three end-to-end case studies. Each case spans the four canonical stages of our pipeline: (i) query specification with method anchors; (ii) typed lineage retrieval from the methodological graph G; (iii) the bottleneck landscape extracted from the evidence records $\rho(\cdot)$; and (iv) the strategy-driven proposal grounded in this evidence. The three cases deliberately span very different regimes—an embodied-AI manipulation task, an AI-for-science query in protein modeling, and a deployment-constrained computer-vision problem—to test whether graph-derived structure transfers beyond mainstream ML methodology.

E.2.1 Sim-to-Real Robot Manipulation

The query specifies a tabletop manipulation goal (*Pour from the blue mug into the green bowl*) and six method anchors (Figure 6) covering language-grounded planning (PALM-E, SAYCAN), end-to-end VLA policies (RT-1, RT-2, OPENVLA), and diffusion-based controllers (DIFFUSION POLICY). Intern-Atlas resolves $28{,}714$ relevant papers and $65{,}605$ typed edges; the lineage panel exposes a clean evolutionary chain via extends, adapts, and compares edges, with OPENVLA reusing pretrained components and DIFFUSION POLICY retained as a parallel comparison baseline representing the visuomotor-policy line. The bottleneck landscape is heavily skewed toward *generalization* ($16{,}492$ signals, 25.1%) and *accuracy* ($7{,}768$), confirming that sim-to-real transfer is the dominant unsolved problem in this lineage. The graph-conditioned generator returns a five-stage agenda—world-model-powered VLA, vision-language-action encoder, morphology tokens with 3D spatial memory, mixed simulation and human-to-robot data, and online self-reflection—in which each module is mapped to a specific bottleneck cluster surfaced by the graph rather than free-text speculation.

E.2.2 AI for Protein

The query consists of generic keywords (*protein design, protein structure, protein LM, AlphaFold, antibody*); Intern-Atlas (Figure 7) anchors retrieval on six landmark methods and surfaces $9{,}347$ candidate papers and $7{,}418$ typed edges. The lineage panel here is structurally distinct from the robotics case: methods do not lie on a single chain but form a hub-and-branch topology in which ALPHAFOLD and ROSETTAFOLD anchor two parallel structure-prediction lines, ESMFOLD compares against ALPHAFOLD on the PLM-driven folding line, RFDIFFUSION extends ROSETTAFOLD into a denoising-based design framework, PROTEINMPNN occupies a separate inverse-folding branch, PROGEN2 improves upon prior sequence-modeling baselines, and PROTEINGYM serves as a comparison benchmark on the PLM branch. The bottleneck distribution is led by *accuracy & physical validity* ($1{,}633$) and *generalization* ($1{,}322$), with *compute & scalability* (924) and *data efficiency* (438) emerging as long-tail concerns characteristic of scientific-discovery domains. The proposed agenda—a multimodal protein foundation model with sequence/3D/surface/ligand encoders,

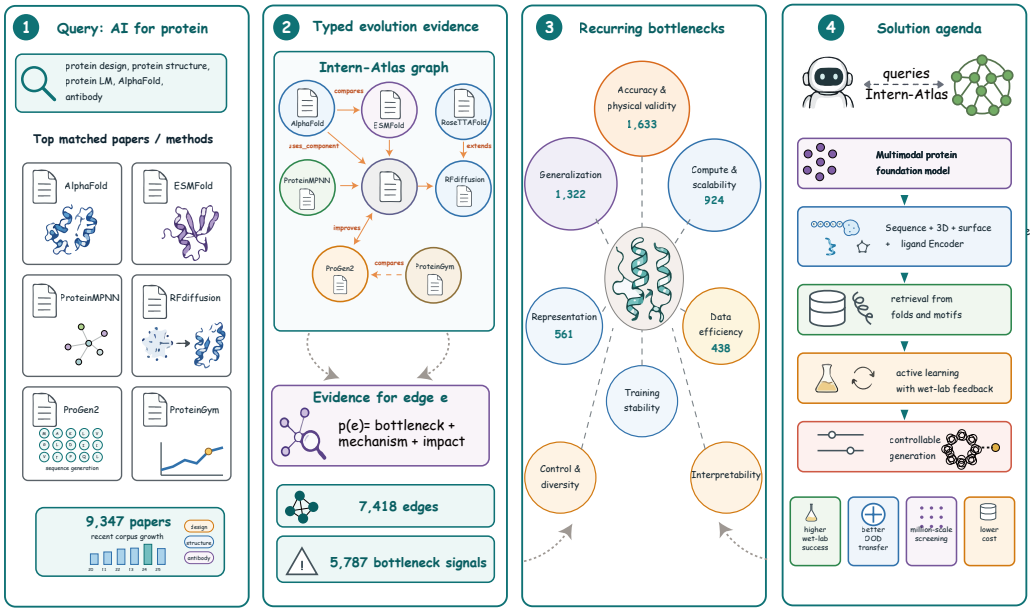


Figure 7: **AI for protein.** A keyword query (protein design, structure, LM, AlphaFold, antibody) over 9,347 in-scope papers anchors retrieval on six landmark methods (ALPHAFOLD, ESMFOLD, PROTEINMPNN, RFDIFFUSION, PROGEN2, PROTEINGYM), with ROSETTAFOLD additionally surfaced as the structural ancestor of RFDIFFUSION via lineage expansion: (1) the query canvas with top matched methods and recent corpus growth across design, structure, and antibody subfields; (2) a typed evolution graph with 7,418 edges in a hub-and-branch topology rather than a single chain, each edge carrying $\rho(e) = (bottleneck, mechanism, impact)$; (3) the recurring-bottleneck distribution (5,787 total signals), led by accuracy & physical validity (1,633) and generalization (1,322); (4) a multimodal protein foundation model with sequence/3D/surface/ligand encoders, fold-and-motif retrieval, active learning with wet-lab feedback, and controllable generation, mapped onto specific impact axes.

fold-and-motif retrieval, and active learning with wet-lab feedback—maps these bottleneck channels directly to architectural commitments, demonstrating that graph-grounded ideation transfers to AI-for-science settings rather than only methodological NLP/ML topics.

E.2.3 Real-Time Open-Vocabulary Object Detection

The third case, shown in Figure 8, stresses a deployment-constrained setting: open-vocabulary detection and segmentation on a Jetson Orin edge device with a < 33 ms/frame latency budget and < 15 W power envelope. Intern-Atlas surfaces 32,584 candidate papers and 289,536 typed edges. Unlike the previous two cases, the lineage panel organizes results into four distinct strata—detector families (YOLO/R-CNN/DETR), Transformer paradigms, open-vocabulary vision, and real-time systems—each with its own backbone choices (RESNET/SWIN/VIT) and dataset/tool resources (COCO, LVIS, OBJECTS365, CITYSCAPES). A representative evidence card is fully expanded, showing how the $\rho(e)$ schema captures the closed-set bias bottleneck, the open-vocabulary mechanism that resolves it, and the quantitative impact ($+7.3$ mAP novel class, $+12.1$ recall on long-tail). The bottleneck landscape is by far the largest of the three cases (59,227 accuracy plus 33,071 generalization signals) and is grounded in concrete failure cases (missed ambulance, false-positive traffic light, person→pole class confusion) shown alongside the bottleneck histogram. The five steps of the proposal are each annotated with the specific impact axis it targets, showing that graph-grounded ideation can respect rather than ignore hardware-level constraints.

Discussion. Across the three cases, the shape of the lineage subgraph differs sharply (chain, hub-and-branch, multi-tier strata), as does the dominant bottleneck profile. Yet the same four-stage operator pipeline returns a coherent, evidence-anchored proposal in every domain, supporting the

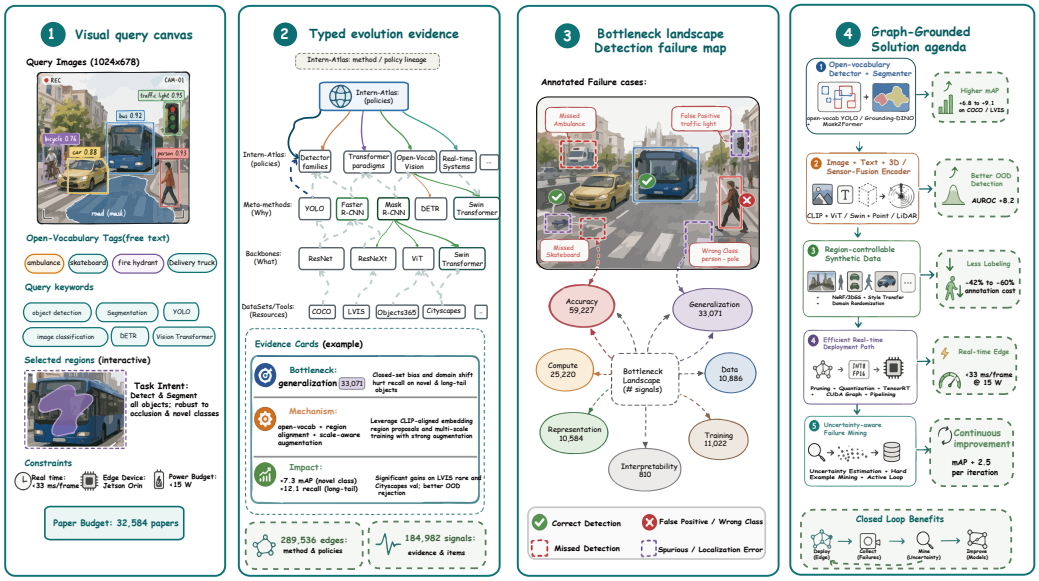


Figure 8: **Real-time open-vocabulary detection on edge devices.** A visual query with hard deployment constraints ($<$ 33 ms/frame, $<$ 15 W on Jetson Orin) drives Intern-Atlas to retrieve 32,584 candidate papers: **(1)** the query canvas with open-vocabulary tags, selected regions, task intent, and resource constraints; **(2)** a four-tier method lineage—policies, meta-methods, backbones, datasets/tools—over 289,536 typed edges, with a fully expanded evidence card showing the $\rho(e)$ schema in action (+7.3 mAP novel class, +12.1 recall on long-tail); **(3)** the bottleneck landscape (*accuracy* 59,227, *generalization* 33,071, *compute* 25,220) overlaid on raw failure cases (missed ambulance, false-positive traffic light, person→pole class confusion); **(4)** a closed-loop five-stage solution agenda with each stage annotated by its target impact.

claim that a single methodological evolution graph can serve as the data layer for ideation across the AI methodological landscape.

F Limitations and Broader Impact

Extraction. Phase-1 edge-type classification accuracy ranges from 70.4% (production model) to 93.0% (audit model), reflecting the inherent difficulty of distinguishing fine-grained causal relations such as *extends* versus *improves*. An incorrect label can change which links the lineage search follows or how it ranks the paths it retrieves. The bottleneck taxonomy $\mathcal{D}$ has 14 fixed axes, and emerging dimensions are mapped to the closest existing axis until a future taxonomy revision. Substring-based alias resolution favors precision over coverage and uses a manually maintained negative list to handle truly ambiguous names.

Updating evidence. When a later paper challenges an earlier claim, the graph adds the new evidence with its source and date. The earlier edge remains traversable, so a search can still return a path that relies on a disputed claim. Recording which paper challenged or superseded a claim would let users filter paths by status and date while retaining the original record.

Algorithmic. The temporal coherence function TC is calibrated on post-2015 AI literature, so fields with markedly different research cadences would require recalibration. The evaluator's zero-trainable-parameter design is a deliberate commitment to deterministic reproducibility, trading potential accuracy gains from a learned scorer for full auditability. The four generation strategies cover the most frequent topological moves we observed; less common modes such as theoretical unification of two previously distinct frameworks are natural extensions.

Evaluating generated ideas. The generator and the automatic evaluator use the same graph, which may favor proposals built around relationships already represented in Intern-Atlas. The blind comparisons in Section 4.3 provide a separate measure of expert preference. An independent execution-based benchmark could test whether the proposed mechanisms resolve their target bottlenecks when implemented.

Broader impact. Agents that use the graph to assign credit may amplify existing citation biases across research communities. The source passages attached to non-background relations let readers check individual attributions. Ranking researchers or institutions falls outside the intended use of the graph.

Data sources and licenses. OpenAlex releases its bibliographic metadata under CC0 1.0. The Semantic Scholar retrieval baseline uses the Academic Graph API under its API License Agreement. Papers obtained from conference proceedings, OpenReview, and arXiv remain subject to their individual publication licenses. The Survey30 release identifies the 30 source surveys and distributes our graph and chain annotations under CC BY-NC 4.0.

The Intern-Atlas code and the Nougat parser code use the MIT License. The Nougat model weights use CC BY-NC 4.0. GROBID 0.8.0 and the Qwen3.6-35B-A3B and Qwen3.5-397B-A17B checkpoints use Apache 2.0. The hosted Claude and GPT models are subject to the applicable service terms from Anthropic and OpenAI.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We clearly state our claims, including the limitations of existing document-centric infrastructure and our contributions with Intern-Atlas, in the Abstract and Introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our LLM-based extraction pipeline and the coverage of the methodological graph in appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This paper focuses on the empirical construction of a methodological evolution graph and tree search algorithms, and does not claim new theoretical bounds.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We describe all experimental settings in both the main paper and the appendix. We detail the graph construction pipeline and the SGT-MCTS algorithm in Section 3, with comprehensive evaluation protocols and hyperparameters provided in Section 4, Appendix B and D.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Sections 3–4 and Appendices A–D describe graph construction and evaluation. Appendix F links to the Survey30 annotations and the software license. Reproducing all main results also requires the experiment inputs and run instructions.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We describe all experimental settings in both the main paper and the appendix. Key implementation details such as the details of the benchmark construction, LLM extraction configurations, and MCTS parameters are detailed to support result interpretation.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the sample mean and standard deviations for score distributions across strata in Figure 4, and compute Spearman correlations for human alignment in Figure 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We detail the exact hardware requirements alongside our experimental results. The large-scale data processing (1M+ papers) was executed using the Qwen3.6-35B-A3B model on a cluster of 48 H200 GPUs (configured as 6 nodes with 8 GPUs each).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: We discuss the broader impacts of accelerating automated scientific discovery, including potential risks related to AI-generated research, in Appendix F.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: Intern-Atlas records relationships among published research methods and attributes them to the source papers.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Appendix F lists the licenses and terms for data sources, parsers, and models. The Survey30 release identifies the original surveys.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We introduce the Intern-Atlas graph and the Strata Dataset. Comprehensive documentation, including dataset structure and collection methods, is provided in the Appendix.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: We engaged 10 AI PhD researchers for expert evaluation (Sections 4.2 and 4.3) and provided appropriate compensation. Prior to the evaluation, all participants were informed of our privacy protocols and provided informed consent regarding the experimental setup and compensation arrangements. As explicitly requested by the subjects, their personal identities are kept strictly confidential and fully anonymized in this paper.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [Yes]

Justification: The human evaluation exclusively involved expert researchers judging scientific ideas based on public academic data, which does not constitute regulated human subjects research requiring formal IRB approval at our institution.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We explicitly describe the use of LLMs for edge typing and evidence extraction during graph construction (Section 3.2), and as an LLM-as-Judge evaluation baseline (Section 4.2).

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.