FORMING FEASIBLE RESEARCH IDEAS BY ANALOGICAL TRANSFER OVER RATIONALE-BEARING GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models are increasingly used to generate research ideas, yet human studies find their ideas weakest on feasibility. Existing work scores or tests feasibility once an idea exists; how the idea takes shape has received little attention. We present `IdeaStrata`, a harness that addresses feasibility while the idea is formed. Cognitive theories of analogy ask of a sound transfer whether a relation, with the conditions under which it holds, remains valid in a new setting, the question on which feasibility turns; `IdeaStrata` makes this check the representation the model operates on and the steps it carries out. Each input paper becomes a Hierarchical Layered Graph whose edges carry a *rationale*, the mechanism and conditions under which a relation holds; rationale transfer maps a relation onto concepts of another paper only when those conditions still hold, and structural verification removes transferred relations that cannot hold together. An idea is written only from relations that pass both checks. On IdeaBench with two backbones, `IdeaStrata` generates the most feasible ideas, with a mean feasibility $1.8\times$ (GPT-4o) and $2.4\times$ (Qwen3.8-Flash) that of the strongest baseline, and shifts the whole feasibility distribution upward, including target papers for which all three generated ideas are judged more feasible than the target itself. Its ideas are least often criticized for the concerns the design targets, such as a relation applied where its conditions do not hold, and removing either the rationale or structural verification lowers feasibility and raises these concerns.

1 INTRODUCTION

Automated scientific idea generation using large language models (LLMs) has emerged as a rapidly growing research direction. Existing methods span multiple technical paradigms: literature-grounded generation (Wang et al., 2024; Baek et al., 2025), multi-agent collaboration (Su et al., 2025; Gottweis et al., 2026), iterative search and evolutionary optimization (Li et al., 2025; Hu et al., 2024; Yu et al., 2026), and end-to-end autonomous systems (Lu et al., 2026; Yuan et al., 2025). These approaches have demonstrated impressive progress in generating ideas that appear promising.

Despite this progress, human studies suggest that feasibility remains a weak point of generated ideas. Si et al. (2025) find that LLM-generated ideas are judged slightly weaker on feasibility than expert-written ones in a blind review involving 49 expert idea authors and 79 expert reviewers, who point to vague implementation details, unrealistic assumptions, and excessive resource demands. Their follow-up study (Si et al., 2026) confirms this through execution: after 43 experts implemented the ideas, the scores of LLM-generated ideas dropped more than those of expert-written ones, and on many metrics the ranking flipped—ideas that look promising on paper do not always hold up once executed. However, only a few works explicitly target feasibility, and they mostly treat it as a property to be scored or tested once an idea exists: Li et al. (2024) add feasibility to the training objective, Vasu et al. (2025) train a reward model to predict execution success, and Yuan et al. (2025) implement ideas in code and feed the results back into the next round. While these efforts have improved how feasibility is evaluated and selected for, the process through which an idea takes shape has received comparatively little attention.

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

The process by which a model arrives at its output is determined not only by the model but also by the harness around it. An agent harness is the non-parametric execution and orchestration layer surrounding a foundation model: it determines how observations are represented, which actions are available, how model outputs are validated, and when execution continues, retries, or terminates. In industrial practice, harness design has become a key lever for making agents reliable on long-running tasks (Young, 2025; Rajasekaran, 2026), complementary to improving the model itself. Cognitive science offers an account of why this lever works: representational determinism holds that the form of a representation determines which operations can be performed on it (Zhang, 1997; Kirsh, 2010), so fixing what the model operates on also fixes which checks it can carry out. For idea generation, this suggests that the process through which an idea takes shape can be governed by designing what the model operates on and in what order, rather than by retraining or scoring after the fact.

To govern this process toward feasibility, we turn to analogy, a core mechanism of scientific discovery (Dunbar, 1997; Dunbar & Blanchette, 2001), and specifically to cognitive theories of what makes an analogical transfer sound. Structure-mapping theory specifies *what* is transferred: relational structure, the relations together with the higher-order constraints that bind them, rather than surface attributes (Gentner, 1983). Multi-constraint theory specifies *when* a mapping is admissible: when it jointly satisfies the constraints of similarity, structure, and purpose (Holyoak & Thagard, 1995; 1997). The Structure-Mapping Engine specifies *how* local mappings are combined: locally established correspondences are jointly evaluated for global consistency (Falkenhainer et al., 1989). Each of these theories asks whether a relation, together with the conditions under which it holds, remains valid in a new setting. This is the question on which the feasibility of an idea turns: an idea built on a relation whose conditions are not met in the target setting rests on an unrealistic assumption.

Building on these insights, we propose `IdeaStrata`, a harness that grounds idea generation in the cognitive science of analogy. The core idea is to integrate these checks into the process of forming an idea (Figure 1): the conditions under which a relation holds become explicit objects on which the model operates, and each check becomes a distinct step invoked by the harness. On the representation side, `IdeaStrata` externalizes the relational structure of the input papers as *Hierarchical Layered Graphs* (HLGs), whose edges carry a *rationale* that captures both the mechanism underlying each relation and the conditions under which it holds. On the process side, it applies three operators to these graphs (Figure 2): *reasoning skill extraction* constructs the HLGs; *rationale transfer* abstracts a source edge into a schema (Gick & Holyoak, 1983) and maps it onto target concepts only when all three constraints are satisfied and the skill's conditions are met; and *structural verification* jointly evaluates the transferred edges for global consistency. The LLM performs every

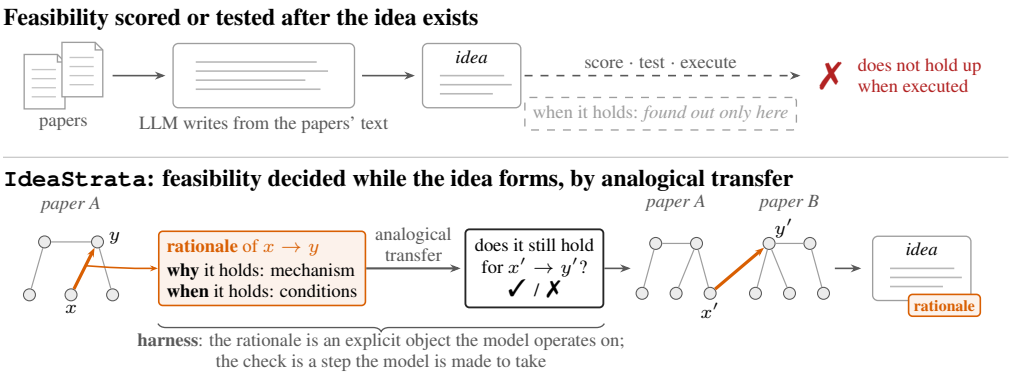

Figure 1: **Where feasibility is decided.** *Top:* existing approaches treat feasibility as a property to be scored or tested once an idea exists, so whether the conditions under which the idea works are met is found out only afterwards. *Bottom:* `IdeaStrata` represents each paper as a graph whose relations carry a *rationale*, why the relation holds and the conditions under which it holds, and forms an idea by analogical transfer: a relation found in paper A licenses a new relation between a concept of paper A and a concept of paper B only if its rationale still holds for them. The check is a step the harness makes the model take before the idea is written, and the idea is the structure that spans the two papers (§3).

operation, and the harness fixes what it operates on and in what order. An idea is verbalized only from relations that have passed both checks, so feasibility is addressed while the idea is formed rather than scored after it exists.

We evaluate `IdeaStrata` on IdeaBench (Guo et al., 2025) against direct LLM generation, VirSci (Su et al., 2025), and AI Scientist-v2 (Yamada et al., 2025), with GPT-4o and Qwen3.8-Flash as backbones. On both backbones, `IdeaStrata` generates the most feasible ideas, and it increases not only the share of target papers for which at least one idea is judged more feasible than the target, but also the share for which two or all three are (Section 4.2). To see where the gain comes from, we tag the concerns raised in the judge's feasibility comments (on Qwen3.8-Flash): ideas from `IdeaStrata` are least often criticized for the concerns its design targets, namely a relation applied where its conditions do not hold, an implausible mechanism, and components insufficient to support the claim (Section 4.3). Ablations trace this to the two components that carry these checks: removing either the rationale or structural verification lowers feasibility and raises these concerns (Section 4.4).

Our contributions are as follows:

- **Paradigm.** We address feasibility in how an idea is formed rather than by scoring it afterwards: through a harness, we turn what analogy theory requires of a sound transfer—what is transferred, when a mapping is admissible, and how mappings combine—into the representation the model operates on and the steps it carries out.
- **Method.** We build `IdeaStrata`, which represents papers as HLGs with rationale-bearing edges and runs analogical transfer as three operators over them, with admissibility checked against the constraints of multi-constraint theory and the result verified for global consistency.
- **Performance.** On IdeaBench, 13.0% of the ideas generated by `IdeaStrata` with Qwen3.8-Flash are judged more feasible than the target paper, against 5.4% for direct generation, the strongest baseline, a $2.4\times$ higher mean feasibility; with GPT-4o, the shares are 21.4% and 12.1%, a $1.8\times$ improvement. A feasibility concern analysis shows that it reduces the concerns its design targets, and ablations show that removing rationale or structural verification lowers feasibility and raises these concerns.

2 Related Work

2.1 LLM-Based Research Ideation and the Feasibility Gap

Recent work on LLM-driven scientific ideation spans several technical paradigms: literature-grounded generation that retrieves and recombines published findings (Wang et al., 2024; Baek et al., 2025), multi-agent collaboration that simulates research-team dynamics (Su et al., 2025; Gottweis et al., 2026), iterative search and evolutionary optimization over idea spaces (Li et al., 2025; Hu et al., 2024; Yu et al., 2026), and end-to-end autonomous systems that close the loop from ideation through execution (Lu et al., 2026; Yuan et al., 2025). These methods have made substantial progress in generating ideas that appear promising.

Human studies nevertheless point to feasibility as a weak point of generated ideas (§1): in a blind review by expert authors and reviewers, Si et al. (2025) find LLM-generated ideas judged slightly weaker on feasibility than expert-written ones, and their follow-up execution study (Si et al., 2026) shows that, once experts had implemented the ideas, the scores of LLM-generated ideas dropped more than those of expert-written ones, with the ranking flipping on many metrics. Several efforts address feasibility directly: Li et al. (2024) add feasibility to the training objective, Vasu et al. (2025) train a reward model to predict execution success, Yuan et al. (2025) implement ideas in code and feed the results back into the next round, and Duan et al. (2025) simulate experimental verification through multi-agent interaction. These approaches improve how feasibility is optimized, scored, or tested once an idea exists; none of them changes how the idea is formed. `IdeaStrata` addresses this complementary dimension and could in principle be combined with reward-based scoring or execution-based filtering.

2.2 Analogical Reasoning and Computational Analogy

The theories on which `IdeaStrata` rests, structure-mapping theory (Gentner, 1983), multi-constraint theory (Holyoak & Thagard, 1995; 1997), and the Structure-Mapping Engine (Falkenhainer et al., 1989), are introduced in §1. Studies of working scientists show that scientific innovation relies on structural rather than surface-level analogies (Dunbar, 1997; Dunbar & Blanchette, 2001), and Gick & Holyoak (1983) show that a schema induced by comparing analogous cases facilitates transfer to new problems.

Webb et al. (2023) show that LLMs solve zero-shot analogy tasks, including matrix reasoning, letter-string transformations, and story analogies, at or above human level. These tasks present the relational structure explicitly. Scientific ideation instead requires extracting relational structure from unstructured literature, transferring it across domains, and verifying the coherence of the result; `IdeaStrata` bridges this gap between possessing the capacity and deploying it in open-ended ideation by fixing what the model operates on and in what order.

Prior work on computational analogy has focused on *analogy retrieval*, that is, finding structurally similar sources. Hope et al. (2017) learn purpose–mechanism representations of product descriptions to mine analogies at scale, and SOLVENT (Chan et al., 2018) extends this to scientific papers with human-annotated background, purpose, mechanism, and findings. Both optimize their representations for matching. `IdeaStrata` addresses the downstream problem of *analogy execution*: given a source, abstracting a skill into a schema, checking whether a mapping is admissible, and verifying the transferred edges for global consistency, so that an analogy becomes a feasible idea.

Finally, HLG draws on a tradition of representing scientific text as structured objects: argumentative zoning decomposes papers by rhetorical function (Teufel & Moens, 2002), SciERC extracts typed entities and relations (Luan et al., 2018), and the Open Research Knowledge Graph represents scholarly knowledge as semantic graphs (Jaradeh et al., 2019). These representations serve information extraction and retrieval; HLG is instead built for analogical transfer, and its distinctive element is the rationale carried by each edge, which records the mechanism behind a relation and the conditions under which it holds.

3 Method

3.1 Overview

`IdeaStrata` treats idea generation as analogical transfer carried out on an explicit representation of the input papers rather than on their text. The design rests on one decision about representation: the conditions under which a relation holds are recorded as objects attached to the relation. Whether those conditions are met in a new setting is the question on which the feasibility of an idea turns (§1); recording them makes that check a comparison between represented objects, one that can be invoked, inspected, and revisited on its own. This is an instance of representational determinism: the form of a representation determines what information can be perceived and what processes can be activated on it (Zhang, 1997), and external representations give "access to new operators—you can do something outside that you cannot inside" (Kirsh, 2010). `IdeaStrata` accordingly externalizes the relational structure of each input paper as a *Hierarchical Layered Graph* (HLG) whose edges carry rationales, and decomposes idea generation into three operators that act on HLG elements (Figure 2). The LLM performs every operation; the harness fixes which objects each operation reads and writes and the order in which the operations run. Apart from building the HLGs, no operator reads the papers' text.

Objects. The HLG of paper d_i is a graph $G_i = (V_i, E_i, \ell_i)$. Its nodes V_i are the scientific concepts of the paper, and $\ell_i : V_i \to \{L_1, L_2, L_3\}$ places each at one of three levels of abstraction: research themes, composable concepts, and operational components (§3.2). Each edge is a tuple $e = (u, v, r, \tau)$, where $u, v \in V_i$, $r \in \mathcal{R}$ is a relation type from a fixed set (Appendix A.1), and τ is a natural-language *rationale* that records the mechanism by which the relation holds and the conditions under which it holds. We call each such edge a *reasoning skill*: through its rationale it captures domain-specific know-how about how and why two concepts in a field are connected, know-how that can be applied to new concepts through analogical transfer; the term denotes this transferable know-how, not a procedure an agent can call. One derived object arises during reasoning: a *latent*

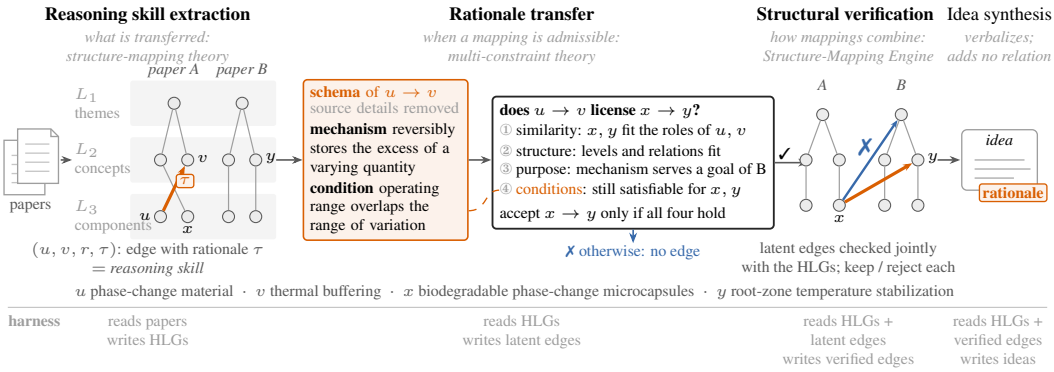

Figure 2: **Overview of `IdeaStrata`.** Each paper is externalized as a HLG whose nodes sit at one of three levels and whose edges carry a *rationale* (orange): the mechanism of the relation and the conditions under which it holds. Three operators act on HLG elements in sequence: reasoning skill extraction builds the HLGs; rationale transfer states a source skill as a schema and admits a latent edge $x \to y$, with its own rationale, only when similarity, structure, purpose, and the skill's conditions all hold for the pair; structural verification checks the latent edges jointly and removes those inconsistent with the rest (blue).

edge is a rationale-bearing edge proposed between concepts of different papers, whose rationale explains why the mechanism holds in the new configuration.

Operators. `IdeaStrata` applies three operators in sequence, each reading the objects the previous one wrote. *Reasoning skill extraction* (§3.2) builds an HLG from each input paper. *Rationale transfer* (§3.3) proposes candidate node pairs across HLGs and decides, for each pair, whether a source skill licenses a latent edge between its two nodes. *Structural verification* (§3.4) examines the latent edges jointly and removes those that are inconsistent with the rest. The surviving edges, together with the HLGs, form the reasoning structure of a candidate idea, which *idea synthesis* (§3.5) verbalizes; synthesis adds no relations of its own.

3.2 Reasoning Skill Extraction

Reasoning skill extraction converts each input paper d_i into its HLG G_i. It is the only operator that reads a paper's text, and every later operator works on what it produces, so what it must record is fixed by what analogical transfer needs. Structure-mapping theory holds that a sound analogy transfers relational structure, that is, relations together with the higher-order constraints that bind them, rather than surface attributes (Gentner, 1983); a constraint that is not represented cannot be mapped. Extraction therefore records, for every relation it extracts, the mechanism by which the relation holds and the conditions under which it holds, as the rationale τ of the edge; a relation label alone would carry the relation and leave its constraints behind. Extraction also places every concept at an explicit level of abstraction, so that concepts from different papers can later be compared by the role they occupy in their papers rather than by their wording.

The three levels distinguish concepts by the role they play in a paper: *research themes* (L_1) are the scientific directions or objectives a paper pursues, *composable concepts* (L_2) are reusable ideas that can take part in cross-paper transfer, and *operational components* (L_3) are the concrete mechanisms, algorithms, models, or system components through which higher-level concepts are realized. Extraction runs once per paper, independently and under the same level definitions: given the text of d_i, the model identifies its concepts, assigns each a level, and records the relations among them as rationale-bearing edges, which we write as $\mathcal{M}(d_i) = G_i$. For instance, a paper may contain the operational component $u =$ *phase-change material* and the composable concept $v =$ *thermal buffering*, joined by the edge $e = (u, v, \textsc{Implements}, \tau)$, where τ states that the material absorbs heat as the temperature rises and releases it as the temperature falls, thereby damping temperature variation, and that this holds only when the material's transition range overlaps the temperature range of the environment. The relation type says *what* connection holds between u and v; the rationale says *why*

it holds and under what conditions the same reasoning may hold elsewhere, which is what makes the edge a reasoning skill that transfer can reuse.

3.3 RATIONALE TRANSFER

Rationale transfer proposes relations that no single input paper contains. Given the HLGs, the model first proposes candidate pairs (x, y) with $x \in V_i$, $y \in V_j$, $i \neq j$, that could take part in an analogy, and for each pair selects a source skill $e = (u, v, r, \tau)$ from the HLGs whose reasoning might apply to it; it then decides whether e licenses a latent edge between x and y. The decision is made on relational structure and the conditions recorded in the rationale, rather than on surface resemblance alone. Two results from the theory of analogy shape how it is made. First, transfer is facilitated when the relational pattern shared by analogous cases is represented apart from the details of the case it came from (Gick & Holyoak, 1983); the model is therefore instructed to first state the skill as a *schema*, its mechanism and conditions with the source-specific details removed, and only then to judge the pair, so that what is tested against the target is the reasoning pattern rather than the wording of the source. This abstraction has content only because the edge carries a rationale: an edge that records only a relation label abstracts to a vacuous schema. Second, multi-constraint theory holds that a mapping is admissible only when it jointly satisfies *similarity*, *structure*, and *purpose* (Holyoak & Thagard, 1995; 1997). Here, similarity is whether the content of x and y fits the roles of the skill's endpoints, structure is whether their levels and the relations they take part in fit those roles, and purpose is whether the transferred mechanism serves a scientific objective in the new configuration. The same judgment also asks whether the conditions recorded in the rationale remain satisfiable for (x, y); this is the check on which feasibility turns (§1), and an unmet condition is caught here, at the level of a single relation. The questions are answered jointly rather than scored separately, and a transfer is accepted only when all hold.

We write the operator as

$$\mathcal{T}(e, x, y; \mathcal{G}) = \begin{cases} (x, y, r^*, \tau^*) & \text{if the judgment accepts,} \\ \varnothing & \text{otherwise,} \end{cases}$$

where the latent edge carries a relation type r^* and a new rationale τ^* that states why the mechanism holds between x and y; E_{latent} denotes the set of all accepted edges. Continuing the example, let $x =$ *biodegradable phase-change microcapsules* (L_3, from one paper) and $y =$ *root-zone temperature stabilization* (L_2, from another) be a candidate pair, and let e be the skill above. Its schema is that a component stabilizes a varying quantity by reversibly storing an excess of it and returning it when the quantity falls, provided the component's operating range overlaps the range of variation. The microcapsules can fill the role of the storing component, the pair preserves the levels of the source endpoints, passive temperature stabilization serves the objective of the receiving paper, and the transition range of such capsules can be chosen to lie within root-zone temperatures; the transfer is accepted, with τ^* stating that the capsules absorb heat in warmer periods and release it in cooler ones, damping root-zone fluctuations. For $y' =$ *soil-moisture stabilization*, the pair looks similar on the surface, but the mechanism stores heat, not water, so the transfer is rejected.

3.4 STRUCTURAL VERIFICATION

Structural verification takes the set E_{latent} of latent edges produced by transfer and returns the subset E_{verified} that can coexist as one reasoning structure. Transfer judges each latent edge on its own, and edges that are valid one at a time may still be inconsistent with each other: two of them may assign the same concept incompatible roles, or rest on mechanisms that contradict one another, or one may depart from the source skill it was derived from. Such conflicts are visible only when the edges are examined together. The Structure-Mapping Engine likewise assembles a globally consistent mapping only after local matching (Falkenhainer et al., 1989). Verification accordingly runs after transfer and examines all latent edges jointly with the HLGs, so that conditions that each relation meets alone but not jointly are caught here.

The operator $\mathcal{V}(E_{\text{latent}}; \mathcal{G}) = E_{\text{verified}} \subseteq E_{\text{latent}}$ is a single guided call: the model receives every latent edge with its rationale, together with the HLGs, and returns a keep-or-reject decision for each edge; it removes edges that are structurally or mechanistically inconsistent with the rest and adds none. In the example, suppose a second latent edge, transferred from a different skill, casts the same

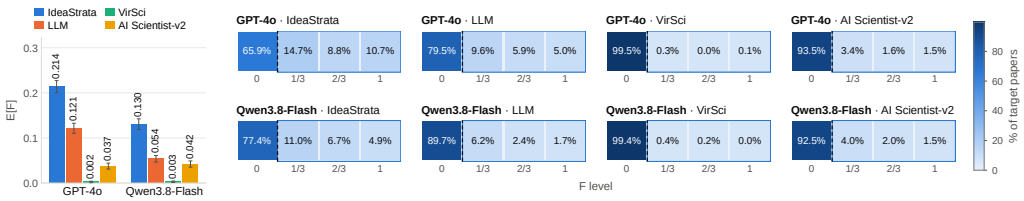

Figure 3: Feasibility of generated ideas, where F is the fraction of the three ideas judged more feasible than the target paper. **Left:** distribution of F over target papers, one panel per backbone (rows) and method (columns); each cell gives the share of target papers at that level, and the dashed line separates $F = 0$ from $F > 0$. **Right:** $\mathbb{E}[F]$ with 95% bootstrap confidence intervals. On both backbones, `IdeaStrata` has the smallest share of target papers at $F = 0$ and the highest $\mathbb{E}[F]$, and its intervals do not overlap with those of any baseline.

microcapsules as a carrier for controlled nutrient release, with a rationale that requires the capsule wall to degrade within days. Each edge is plausible on its own, but the first requires the capsules to persist through repeated thermal cycles over a growing season while the second requires them to break down; the two mechanisms cannot hold of the same component at once, and verification removes the edge that conflicts with the rest of the structure.

3.5 Idea Synthesis: Verbalizing a Verified Reasoning Structure

Synthesis reads the HLGs together with the verified edges, $\mathcal{G} \cup E_{\text{verified}}$, and writes m research ideas ($m = 3$ by default), $\{I_1, \ldots, I_m\} = \text{Synthesize}(\mathcal{G}, E_{\text{verified}}; m)$. It is the only operator whose output is free text, and it reads the graph rather than the papers: the concepts, the relations among them, and the rationale behind every relation, transferred ones included, are in front of the model when it writes, so it composes an idea from relations whose conditions have been checked rather than discovering a reasoning path while writing. When no latent edge survives verification, synthesis runs on the input HLGs alone, and the resulting ideas contain no cross-paper relation. The three operators thus place the two checks that §1 draws from analogy theory before the idea is written: transfer asks whether a relation's conditions are met for one pair, verification whether the retained relations can hold together, and synthesis writes only from what passed.

4 Experiments

4.1 Experimental Setup

Benchmark and metric. We evaluate on IdeaBench (Guo et al., 2025), which contains $2,374$ target papers, each paired with its reference list. Following the low-resource setting of IdeaBench, a method receives, for each target paper, the five most-cited papers in its reference list, with the target itself held out, and generates three ideas. A judge model (GPT-5.6-sol) then ranks the three ideas together with the target paper by feasibility. We score each target paper by the fraction of generated ideas ranked above it, $F \in \{0, \frac{1}{3}, \frac{2}{3}, 1\}$, which is the feasibility insight score of IdeaBench, and report both the distribution of F over target papers and its mean $\mathbb{E}[F]$.

Baselines and backbones. We compare `IdeaStrata` with three baselines: direct LLM generation, which prompts the backbone with the same references to produce three ideas; VirSci (Su et al., 2025), a multi-agent system that simulates a team of scientists; and AI Scientist-v2 (Yamada et al., 2025), of which we run only the idea-generation stage. All methods share the same inputs and are run on two backbones, GPT-4o and Qwen3.8-Flash, with temperature 0. Implementation details are given in Appendix A.2.

4.2 Main Results

`IdeaStrata` generates more feasible ideas. Figure 3 compares the four methods on both backbones. `IdeaStrata` attains the highest mean feasibility, $\mathbb{E}[F] = 0.214$ on GPT-4o and 0.130 on

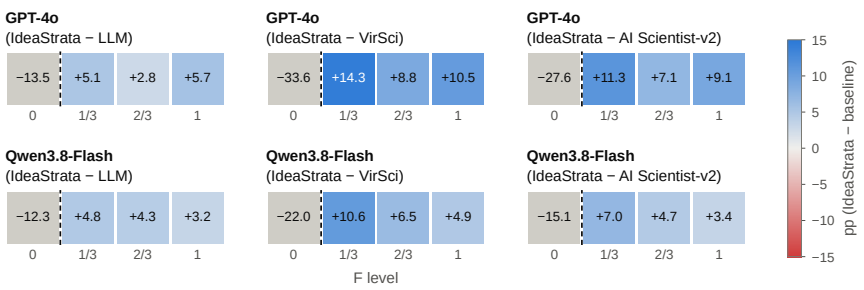

Figure 4: Improvement of IdeaStrata over each baseline, per feasibility level. Each cell gives the difference in the share of target papers at that level of F (IdeaStrata minus baseline, in percentage points); blue means IdeaStrata places more target papers there. The grey $F = 0$ column is the negative sum of the other three. Against every baseline and on both backbones, IdeaStrata moves target papers from $F = 0$ to every positive level, including $F = 1$.

Qwen3.8-Flash, against 0.121 and 0.054 for direct LLM generation, the strongest baseline. The distributions show the same ordering: IdeaStrata leaves the smallest share of target papers at $F = 0$ (65.9% and 77.4%).

IdeaStrata shifts the whole feasibility distribution upward. In Figure 3 (left), IdeaStrata has the largest share of target papers at every positive level of F on both backbones. Figure 4 makes the comparison explicit: against every baseline, IdeaStrata moves target papers out of $F = 0$ into each of $F = \frac{1}{3}$, $\frac{2}{3}$, and 1. Whatever threshold on F is used, a larger share of target papers reaches it with IdeaStrata than with any baseline, including $F = 1$, where all three generated ideas are judged more feasible than the target paper itself.

4.3 Feasibility Concern Analysis

To see which feasibility problems IdeaStrata reduces, we tag the feasibility comment that the evaluation judge writes for each generated idea when ranking it. A tagging model, identical to the evaluation judge, reads only this comment and assigns one or more of ten concern tags, each marking a concrete problem that the comment points out (definitions and setup in Appendix A.3). We run this analysis on Qwen3.8-Flash and count ideas rather than target papers: each target paper contributes three ideas, and each share is taken over all ideas generated by a method.

We group the tags by what the design of IdeaStrata checks. An idea draws its claim from the relations it builds on. We call a concern a *logic-validity* concern if it questions whether the claim still follows from these relations in the new setting, the question on which feasibility turns (§1). The claim can fail to follow in three places, and each has one tag. A relation may not hold at all, because its mechanism is implausible (Mechanism); it may hold, but not under the conditions of the setting where it is applied (Precondition); or the relations may hold where applied and still fall short of the claim, because the proposed components are not enough to support it (Level gap). These correspond to the mechanism and the conditions recorded in each rationale and to the levels of the HLG, and rationale transfer and structural verification are designed to catch them before the idea is written (Section 3). *Execution-related* pools the tags on whether an idea can be carried out in practice, and *Others* the rest.

IdeaStrata reduces logic-validity concerns. As Table 1 shows, IdeaStrata has the lowest share on all three logic-validity tags, including against direct LLM generation, which uses the same backbone and inputs (e.g., 10.0% vs. 24.7% on Mechanism). IdeaStrata also has the lowest share of execution-related concerns, though not on every tag within the group (Table 6).

4.4 Ablation Study

We ablate the two components that carry the checks of analogy theory. *w/o rationale* keeps only a relation label on each HLG edge, dropping the mechanism and the conditions under which the

Table 1: Share (%) of ideas whose feasibility comment receives each concern tag (Qwen3.8-Flash); an idea can receive several tags. *Execution-related* and *Others* give the share of ideas receiving at least one of their three and four tags, respectively; NO ISSUE means no concern is raised. Bold: best per row. All tags are in Table 6.

Tag	IdeaStrata	LLM	VirSci	AI Scientist-v2
Logic-validity				
PRECONDITION	**5.2**	7.2	6.2	14.6
MECHANISM	**10.0**	24.7	11.6	21.8
LEVEL GAP	**11.4**	14.9	13.8	23.1
Execution-related	**60.5**	72.2	72.1	90.0
Others	7.7	11.4	12.2	**5.7**
NO ISSUE	**4.0**	1.6	1.3	0.5

Table 2: Ablation of IdeaStrata (Qwen3.8-Flash, 10% subset). Tags are shares (%) of ideas, as in Table 1. Parentheses: change relative to the full IdeaStrata (percentage for $\mathbb{E}[F]$, ratio for tags). Values come from a separate run and are not comparable with Figure 3 or Table 1.

	IdeaStrata	w/o verification	w/o rationale
$\mathbb{E}[F] \uparrow$	**0.159**	0.116 (-26.9%)	0.102 (-35.6%)
PRECONDITION $\downarrow$	**7.3**	10.4 ($\times 1.42$)	9.3 ($\times 1.27$)
MECHANISM $\downarrow$	**14.2**	20.0 ($\times 1.41$)	18.2 ($\times 1.28$)
LEVEL GAP $\downarrow$	**11.5**	17.3 ($\times 1.51$)	16.5 ($\times 1.44$)

relation holds; *w/o verification* generates ideas directly after rationale transfer, skipping structural verification. Both are run with Qwen3.8-Flash on a random 10% subset of target papers, together with a fresh run of the full IdeaStrata on the same subset, and are evaluated with the same judge and concern tagging as above (details in Appendix A.4).

Both components improve feasibility and reduce logic-validity concerns. As Table 2 shows, removing either component lowers $\mathbb{E}[F]$, from 0.159 to 0.116 without verification and to 0.102 without rationale. It also raises all three logic-validity concerns.

5 DISCUSSION

IdeaBench also scores the novelty of generated ideas, but we evaluate IdeaStrata on feasibility alone, because this is where current systems fall short: in the large-scale human study of Si et al. (2025), LLM-generated ideas were already judged more novel than expert-written ones, while falling behind on feasibility. Novelty in IdeaBench is, like feasibility, judged relative to a single target paper, so ranking above that paper does not by itself establish novelty with respect to the wider literature. How to evaluate novelty and feasibility jointly, so that an idea is credited for being both new to the literature and realizable, remains an open question and a natural next step for benchmarks of AI ideation.

6 CONCLUSION

IdeaStrata addresses the feasibility of generated research ideas in how an idea is formed rather than by scoring it afterwards: the conditions under which a relation holds are recorded on the graph the model operates on, and a relation is transferred only when they hold in the new setting and kept only when the transferred relations hold together. On IdeaBench, this yields the most feasible ideas on both backbones and the fewest of the concerns the design targets, and both checks contribute. Because the harness changes how an idea is formed rather than how it is scored, it complements methods that score or execute ideas after the fact.

AI USE STATEMENT

In this work, we used generative AI tools to assist with software implementation, including drafting, debugging, and refining experimental code. All AI-assisted code was manually reviewed and tested by the authors before being used in the experiments. Generative AI tools were also used to assist with drafting and editing portions of the manuscript for clarity and readability, and with the creation and refinement of scientific illustrations.

Generative AI tools were not used to generate experimental data or synthetic datasets, and all reported experimental results were obtained from the actual implementation and execution of the described experiments. The authors independently verified the experimental procedures, outputs, analyses, and claims presented in the paper.

We have reviewed all AI-assisted work and take full responsibility for the final content of this work, including text, code, figures, claims, and other artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We release the full `IdeaStrata` codebase (generation and evaluation scripts), the complete target/reference dataset, and the generation settings used in our experiments (including model choices, decoding hyperparameters, and a fixed sampling seed of 0). Together, these make the pipeline fully runnable end-to-end. Because several stages rely on LLM calls, residual nondeterminism from provider-side sampling and API behavior can still produce small variations across runs even under identical configuration.

REFERENCES

Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. ResearchAgent: Iterative research idea generation over scientific literature with large language models. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2025.

Joel Chan, Joseph Chee Chang, Tom Hope, Dafna Shahaf, and Aniket Kittur. SOLVENT: A mixed-initiative system for finding analogies between research papers. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):31:1–31:21, 2018. doi: 10.1145/3274300.

Shiyang Duan, Yuan Tian, Qi Bing, and Xiaowei Shao. Bayes-entropy collaborative driven agents for research hypotheses generation and optimization. *arXiv preprint arXiv:2508.01746*, 2025.

Kevin Dunbar. How scientists think: On-line creativity and conceptual change in science. In Thomas B. Ward, Steven M. Smith, and Jyotsna Vaid (eds.), *Creative Thought: An Investigation of Conceptual Structures and Processes*, pp. 461–493. American Psychological Association, 1997.

Kevin Dunbar and Isabelle Blanchette. The in vivo/in vitro approach to cognition: The case of analogy. *Trends in Cognitive Sciences*, 5(8):334–339, 2001. doi: 10.1016/S1364-6613(00)01698-3.

Brian Falkenhainer, Kenneth D. Forbus, and Dedre Gentner. The structure-mapping engine: Algorithm and examples. *Artificial Intelligence*, 41(1):1–63, 1989. doi: 10.1016/0004-3702(89)90077-5.

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

Mary L. Gick and Keith J. Holyoak. Schema induction and analogical transfer. *Cognitive Psychology*, 15(1):1–38, 1983. doi: 10.1016/0010-0285(83)90002-6.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Rory Comanescu, Tao Tu Gabriel Bauer, Yun Shi, et al. Accelerating scientific discovery with AI co-scientist. *Nature*, 2026. doi: 10.1038/s41586-026-10644-y.

Sikun Guo, Haining Cao, Dong Lyu, Zhimin Xiao, and Hao Li. IdeaBench: Benchmarking large language models for research idea generation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2025.

Keith J. Holyoak and Paul Thagard. *Mental Leaps: Analogy in Creative Thought*. MIT Press, 1995.

Keith J. Holyoak and Paul Thagard. The analogical mind. *American Psychologist*, 52(1):35–44, 1997. doi: 10.1037/0003-066X.52.1.35.

Tom Hope, Joel Chan, Aniket Kittur, and Dafna Shahaf. Accelerating innovation through analogy mining. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 235–243. Association for Computing Machinery, 2017. doi: 10.1145/3097983.3098038.

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.

Mohamad Yaser Jaradeh, Allard Oelen, Khedr Elassady Farfar, Manuel Prinz, Jennifer D'Souza, Gábor Kismihók, Markus Stocker, and Sören Auer. Open research knowledge graph: Next generation infrastructure for semantic scholarly knowledge. In *Proceedings of the 10th International Conference on Knowledge Capture (K-CAP)*, pp. 243–246, 2019.

David Kirsh. Thinking with external representations. *AI & Society*, 25(4):441–454, 2010. doi: 10.1007/s00146-010-0272-8.

Long Li, Weiwen Xu, Jiayan Guo, Ruochen Zhao, Xingxuan Li, Yuqian Yuan, Boqiang Zhang, Yuming Jiang, Yifei Xin, Ronghao Dang, Yu Rong, Deli Zhao, Tian Feng, and Lidong Bing. Chain-of-ideas: Revolutionizing research via novel idea development with LLM agents. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025.

Ruochen Li, Liqiang Jing, Chi Han, Jiawei Zhou, and Xinya Du. LDC: Learning to generate research idea with dynamic control. *arXiv preprint arXiv:2412.14626*, 2024.

Chris Lu, Robert Tjarko Lange, Yutaro Yamada, Shengran Hu, Jakob Foerster, David Ha, and Jeff Clune. Towards end-to-end automation of AI research. *Nature*, 651(8107):914–919, 2026. doi: 10.1038/s41586-026-10265-5.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3219–3232, 2018.

Prithvi Rajasekaran. Harness design for long-running application development. `https://www.anthropic.com/engineering/harness-design-long-running-apps`, March 2026. Anthropic. Accessed: 2026-09-26.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. Can LLMs generate novel research ideas? a large-scale human study with 100+ NLP researchers. In *Proceedings of the Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Chenglei Si, Tatsunori Hashimoto, and Diyi Yang. The ideation-execution gap: Execution outcomes of LLM-generated versus human research ideas. In *Proceedings of the Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

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

Simone Teufel and Marc Moens. Summarizing scientific articles: Experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4):409–445, 2002. doi: 10.1162/089120102762671936.

Rosni Vasu, Peter Jansen, Pao Siangliulue, Cristina Sarasua, Abraham Bernstein, Peter Clark, and Bhavana Dalvi Mishra. HARPA: A testability-driven, literature-grounded framework for research ideation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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 (ACL)*, 2024.

Taylor Webb, Keith J. Holyoak, and Hongjing Lu. Emergent analogical reasoning in large language models. *Nature Human Behaviour*, 7(9):1526–1541, 2023. doi: 10.1038/s41562-023-01659-w.

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.

Justin Young. Effective harnesses for long-running agents. `https://www.anthropic.com/engineering/effective-harnesses-for-long-running-agents`, November 2025. Anthropic. Accessed: 2026-09-26.

Jiaxin Yu, Yuxing Song, Zhihui Yan, et al. Large discovery models: Empirically-grounded model-based open-ended search. *arXiv preprint arXiv:2608.15669*, 2026.

Jiakang Yuan, Xiangchao Yan, Bo Zhang, Tao Chen, Botian Shi, Wanli Ouyang, Yu Qiao, Lei Bai, and Bowen Zhou. Dolphin: Moving towards closed-loop auto-research through thinking, practice, and feedback. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

Jiajie Zhang. The nature of external representations in problem solving. *Cognitive Science*, 21(2): 179–217, 1997. doi: 10.1207/s15516709cog2102_3.

A Appendix

A.1 Relation Types

Table 3 lists the fixed relation type set $\mathcal{R}$ from which every HLG edge takes its type r and every latent edge its type r^*. Abstracts and Implements hold only between concepts at different levels; the other three types may also hold between concepts at the same level.

Table 3: The relation type set $\mathcal{R}$. *Scope* states whether a type holds only between concepts at different levels (cross-level) or also between concepts at the same level (any).

Relation type	Scope	Definition
Abstracts	Cross-level	The higher-level concept is an abstraction or generalization of the lower-level concept.
Implements	Cross-level	The lower-level concept is a concrete implementation of the higher-level concept.
Enables	Any	One concept enables or supports the functioning of the other.
Interacts with	Any	The two concepts interact with, or depend on, each other.
Evolves into	Any	One concept evolves or is refined into the other over time.

A.2 Experimental Details

Generation. All methods receive the same inputs for each target paper: the five most-cited papers in its reference list, with the target paper held out. Each method generates three ideas per target paper. Table 4 lists the generation settings shared by `IdeaStrata` and direct LLM generation. For Qwen3.8-Flash we set the reasoning effort to `low`; GPT-4o has no reasoning mode. The maximum output length is left at the provider default for both backbones.

Table 4: Generation settings shared by IdeaStrata and direct LLM generation.

Setting	Value
Backbones	GPT-4o, Qwen3.8-Flash
Ideas per target paper	3
Reference papers per target paper	5 (most cited)
Target paper held out	Yes
Temperature	0
Seed	0
Reasoning effort	low (Qwen3.8-Flash); n/a (GPT-4o)
Maximum output tokens	Provider default

Baselines. *Direct LLM generation* prompts the backbone with the five reference papers and asks for three ideas. *AI Scientist-v2* follows the official implementation[1]; we run only its idea-generation stage and skip the later experimentation and writing stages. We set max-num-generations= 3 and num-reflections= 3, and replace its Semantic Scholar retrieval with the reference papers given by IdeaBench, so that it receives the same inputs as the other methods. *VirSci* follows the official implementation[2] with its default author adjacency matrix, and we set runs= 2, team_limit= 1, and epochs= 3.

Evaluation. For each target paper, the judge model (GPT-5.6-sol) ranks the three generated ideas together with the target paper by feasibility, following the ranking protocol of IdeaBench (Guo et al., 2025). If the target paper is ranked x-th among the four, it receives the score $F = (x-1)/3$, i.e., the fraction of generated ideas ranked above it. The judge uses reasoning effort low and temperature 0 for all methods and backbones.

A.3 Feasibility Concern Tagging Details

Input and model. During evaluation, the judge writes a short feasibility comment for each generated idea when ranking it against the target paper. The tagging model reads only this comment; it does not see the idea itself, the method that produced it, the target paper, or any score, so that it labels the problems raised by the evaluation rather than re-evaluating the idea. The tagging model is the same as the evaluation judge (GPT-5.6-sol, reasoning effort low, temperature 0). We tag every generated idea of the four methods on Qwen3.8-Flash.

Instruction. The tagging model is asked to tag each concrete feasibility problem the comment points out and to skip vague remarks such as "ambitious", "challenging", or "needs careful validation". An idea can receive several tags. A comment that raises no concern is tagged NO ISSUE; a comment that raises only vague concerns receives no tag. For each tag, the model quotes the words of the comment that support it. Table 5 lists the tags with the descriptions given to the model.

Results for all tags. Table 6 reports the shares for all ten tags. IdeaStrata has the lowest share on the three logic-validity tags and on RESOURCE; the reduction does not extend to every tag, and on UNDERSPECIFIED and ABSTRACTION, IdeaStrata is higher than direct LLM generation.

Ideas that do not surpass the target paper. Most generated ideas (94%) are ranked below the target paper. Restricting the analysis to these ideas gives nearly the same shares as Table 6; in particular, IdeaStrata still has the lowest share on all three logic-validity tags.

A.4 Ablation Details

Conditions. All three conditions use Qwen3.8-Flash and the generation settings in Table 4, and are run on the same random 10% subset of IdeaBench target papers. *Full* is a fresh run of IdeaStrata on this subset, so that all three conditions come from runs made under the same setup. *w/o rationale*

[1] https://github.com/sakanaai/ai-scientist-v2

[2] https://github.com/RenqiChen/Virtual-Scientists

Table 5: Concern tags and the descriptions given to the tagging model.

Tag	Description
Logic-validity	
PRECONDITION	A method or finding is applied to a setting where it does not fit.
MECHANISM	The core mechanism or assumption is implausible.
LEVEL GAP	The proposed experiments or components are not enough to support the claim.
Execution-related	
RESOURCE	Needed data, samples, equipment, or techniques are hard to obtain or too costly.
VALIDATION	Confounders, unclear causal attribution, or measurement noise make the claim hard to establish.
UNDERSPECIFIED	A needed step or procedure is left unspecified.
Others	
ABSTRACTION	The idea is too broad or generic to act on.
INCOHERENCE	The parts do not fit together, or contradict each other.
MOTIVATION	The goal or problem itself is not justified or not worth pursuing.
OTHER	A concrete problem not covered above.
NO ISSUE	No feasibility concern.

Table 6: All concern tags assigned to the feasibility comments of generated ideas (Qwen3.8-Flash). Each entry is the share (%) of a method's ideas, three per target paper, whose comment receives the tag; an idea can receive several tags, so columns do not sum to 100%. Tags are grouped into logic-validity concerns, which the design of `IdeaStrata` targets, execution-related concerns, and others; NO ISSUE marks comments that raise no feasibility concern. Bold: lowest share per tag (highest for NO ISSUE).

Tag	`IdeaStrata`	LLM	VirSci	AI Scientist-v2
Logic-validity				
PRECONDITION	**5.2**	7.2	6.2	14.6
MECHANISM	**10.0**	24.7	11.6	21.8
LEVEL GAP	**11.4**	14.9	13.8	23.1
Execution-related				
RESOURCE	**33.0**	41.2	61.8	50.8
VALIDATION	35.0	53.7	**31.1**	76.7
UNDERSPECIFIED	8.4	6.6	**3.5**	9.3
Others				
ABSTRACTION	3.5	3.0	**1.5**	2.2
INCOHERENCE	2.9	5.1	5.6	**1.6**
MOTIVATION	0.2	0.2	0.7	**0.1**
OTHER	**1.4**	3.7	4.9	1.9
NO ISSUE	**4.0**	1.6	1.3	0.5

keeps the HLGs and all operators, but each edge carries only a relation label; the mechanism behind the relation and the conditions under which it holds are removed. *w/o verification* keeps the HLGs and rationale transfer, and generates ideas directly from the transferred edges without structural verification.

Evaluation and statistics. Feasibility scores and concern tags are obtained exactly as in the main experiment (Appendices A.2 and A.3). The three logic-validity tags and NO ISSUE were fixed as the tags of interest before the results were examined; all other tags are reported as secondary. Differences are paired over target papers: for each target paper we take the share of its three ideas

carrying a tag, subtract between conditions, and average over target papers. Confidence intervals are 95% paired bootstrap intervals over target papers ($B = 1000$); p-values come from a two-sided paired permutation test with 10,000 permutations and are not corrected for multiple comparisons.

Full results. Table 7 reports all tags. RESOURCE and VALIDATION do not rise in either ablation: they change by -2.6 and -4.4 points without verification and by $+1.2$ and $+0.2$ points without rationale, none of them significant. The third execution-related tag, UNDERSPECIFIED, does rise, by 3.8 points without verification ($p = 0.006$) and 2.0 points without rationale ($p = 0.20$). The logic-validity tags rise significantly on all three without verification and on LEVEL GAP without rationale ($p < 0.05$). Comparing the two ablations directly, no difference is significant ($p \geq 0.08$ for all tags). Both ablations increase the mean number of tags per idea by about 0.16, so part of the rise in logic-validity tags occurs alongside a general increase in tags; this increase, however, does not extend to the execution-related tags. The median length of the feasibility comments is nearly identical across the three conditions (296–299 characters), so these differences are unlikely to reflect comment length. Restricting the analysis to ideas that do not surpass the target paper leaves the ordering of the three conditions unchanged on all three logic-validity tags.

Table 7: All concern tags in the ablation (share of ideas, %), with the paired difference from the full `IdeaStrata`, its 95% bootstrap confidence interval, and the raw permutation p-value.

	`IdeaStrata`	w/o verification		w/o rationale	
Tag	share	Δ [95% CI]	p	Δ [95% CI]	p
Logic-validity					
PRECONDITION	7.3	+3.1 [+0.5, +5.5]	0.021	+2.0 [−0.6, +4.7]	0.172
MECHANISM	14.2	+5.8 [+2.1, +9.8]	0.003	+4.0 [−0.2, +7.8]	0.051
LEVEL GAP	11.5	+5.8 [+1.8, +9.9]	0.008	+5.0 [+1.2, +8.9]	0.015
Execution-related					
RESOURCE	40.1	−2.6 [−7.2, +1.4]	0.266	+1.2 [−3.2, +6.0]	0.641
VALIDATION	51.4	−4.4 [−9.2, +0.5]	0.088	+0.2 [−4.6, +5.5]	1.000
UNDERSPECIFIED	8.7	+3.8 [+1.2, +6.6]	0.006	+2.0 [−0.6, +4.7]	0.202
Others					
ABSTRACTION	3.1	+0.2 [−1.8, +2.0]	1.000	−0.3 [−2.1, +1.4]	0.873
INCOHERENCE	3.4	+4.1 [+2.0, +6.4]	0.001	+3.1 [+0.8, +5.5]	0.014
MOTIVATION	0.3	+0.2 [−0.5, +0.8]	1.000	+0.3 [−0.3, +1.1]	0.693
OTHER	4.4	+0.0 [−2.0, +2.1]	1.000	−1.1 [−2.8, +0.6]	0.294
NO ISSUE	3.5	+0.9 [−0.9, +2.9]	0.444	−0.3 [−1.8, +1.4]	0.858