When Idea Generation Scales: Idea-Level Evaluation for Autonomous Scientific Discovery

Anonymous ACL submission

Abstract

Large language models are increasingly being built into autonomous research agents that can propose hypotheses, design methods, and draft scientific papers. Yet idea generation is scaling faster than the evaluation signals used to judge it: early-stage ideas rarely have gold answers, experiments, or complete manuscripts. We therefore propose IDEAL, an idea-level method and evaluation framework for measuring sets of generated scientific ideas before downstream validation. IDEAL couples Forge, a pipeline for generating implementation-oriented ideas, with Lens, an evaluation suite that measures internal semantic diversity and external literature-level atypicality. Across 31 scientific tasks, IDEAL improves four idea-level metrics by an average of 82.3% over the strongest non-ours baselines after orienting all metrics so that higher is better.

1 Introduction

Autonomous scientific discovery is becoming a systems problem rather than a distant ambition, with recent agents automating substantial parts of the research loop from ideation to experimentation and manuscript drafting (Lu et al., 2024; Yamada et al., 2025). The scope of these systems is also broadening: research agents now support literature-grounded ideation (Baek et al., 2025), collaborative hypothesis refinement (Gottweis et al., 2025), multi-agent scientific reasoning (Ghafarollahi and Buehler, 2025; Yang et al., 2026), and end-to-end research automation (Liu et al., 2026; Zhou et al., 2025). This trajectory suggests that LLMs are no longer only tools for summarizing papers or assisting with isolated subtasks, but are increasingly used to generate candidate scientific directions (Ren et al., 2025; Zheng et al., 2025a; Wei et al., 2025). This raises the question illustrated in Figure 1: not only whether agents can produce plausible ideas, but whether those ideas are worth scientific attention (Si et al., 2024; Guo et al., 2025).

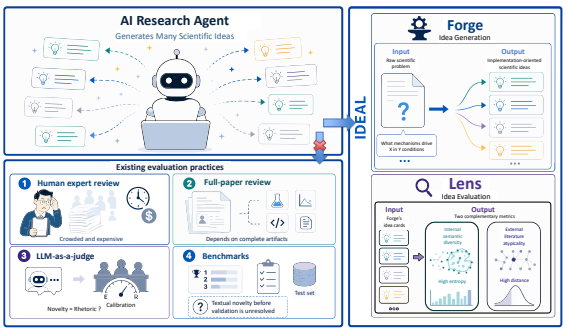

Figure 1: IDEAL adds idea-level generation and evaluation to autonomous research workflows.

This shift changes the unit of evaluation, because idea generation does not fit the same evaluation regime as question answering or benchmark task solving (Sun et al., 2025; Zheng et al., 2025b). When a model answers a question, correctness can often be checked against a reference answer or task metric (Li et al., 2025; Joseph et al., 2025). When a research agent produces many plausible ideas, however, the harder question is whether these ideas are genuinely distinct, novel, and actionable (Tong et al., 2026a; Ye et al., 2026). We therefore treat the evaluation object as a set of early-stage ideas generated for the same scientific problem, before experiments, code, or full manuscripts are available. In this setting, there is usually no gold answer; the evaluator must instead measure whether the set expands the space of candidate scientific directions without collapsing into redundant variants. This makes idea generation a set-level evaluation problem rather than a single-output scoring problem. A fluent proposal may still be scientifically thin: it may rename an existing method, recombine familiar concepts, or restate the same direction under several surface forms (Si et al., 2024; Vasu et al., 2025). This is especially problematic for AutoResearch systems, where a weak evaluation signal can feed back into the research loop and amplify ideas that are polished but redundant (Lu et al., 2024;

Yamada et al., 2025; Baumann et al., 2026). In short, idea generation is beginning to scale, but the evaluation signal has not scaled with it (Xie et al., 2025; Zhang et al., 2025a).

Existing evaluation practices are poorly matched to early-stage scientific ideas (Guo et al., 2025; Si et al., 2024). Human expert review is informative but expensive, domain-dependent, and difficult to scale (Si et al., 2024; Tong et al., 2026a). Full-paper review assumes artifacts such as experiments, figures, code, and manuscripts, which ideas often do not yet have (Zhu et al., 2025; Baumann et al., 2026). LLM-as-a-judge offers a convenient alternative, but recent work suggests that model judgments require careful calibration and may conflate novelty with rhetorical quality or reward exciting but weakly testable proposals (Zhuge et al., 2024; Zhu et al., 2025; Baumann et al., 2026). Benchmarks for scientific reasoning and research agents have broadened the range of tasks used to test scientific capability (Shinde et al., 2026; Tong et al., 2026b; Li et al., 2025; Zheng et al., 2025b), but they do not by themselves resolve how to evaluate the novelty of a textual idea before experimental validation (Sun et al., 2025; Joseph et al., 2025). What remains missing is a scalable way to evaluate a set of early-stage ideas before experiments: whether the set is internally non-redundant and externally uncommon with respect to the literature.

This paper treats scientific idea evaluation as a missing measurement layer for autonomous research agents (Zhang et al., 2025b,a). We present IDEAL, an idea-level framework built around two components: Forge, which generates implementation-oriented scientific ideas from problem descriptions, and Lens, which evaluates ideas through internal semantic diversity and external literature-level atypicality. Lens asks two measurable questions: internally, whether a system produces semantically distinct proposals rather than paraphrases of the same direction, and externally, whether those proposals connect problem-side and method-side concepts in ways that are atypical in the scientific literature (Si et al., 2024; Guo et al., 2025). We evaluate IDEAL with semantic entropy, context-aware diversity, normalized pointwise mutual information, and atypicality scores across weather and climate, wireless communication, remote sensing, and environmental science benchmarks (Shinde et al., 2026; Tong et al., 2026b). This design allows us to compare idea generators not only by how persuasive their outputs sound, but by whether they expand the semantic and conceptual space of candidate scientific directions (Tong et al., 2026a; Ye et al., 2026).

Contributions. We make three contributions:

- We introduce **IDEAL**, an idea-level framework for autonomous scientific discovery that separates internal semantic diversity from external literature-level atypicality.
- We instantiate **Forge**, an AutoResearch pipeline that generates implementation-oriented scientific ideas from problem descriptions, enabling controlled comparisons against existing research-agent systems.
- We develop **Lens**, a multi-domain evaluation suite for scientific ideas, and validate IDEAL across weather and climate, wireless communication, remote sensing, and environmental science benchmarks. Across four aggregate idea-level metrics, IDEAL achieves an average oriented relative improvement of 82.3% over the strongest non-ours baselines.

2 Related Work

Autonomous scientific agents. LLM-driven scientific discovery has rapidly expanded from single-step ideation to agentic workflows that cover literature retrieval, hypothesis generation, experiment execution, paper writing, and review. The AI Scientist line formulates open-ended discovery as an automated loop from idea generation to manuscript production (Lu et al., 2024; Yamada et al., 2025), while AI co-scientist systems emphasize collaborative hypothesis generation and refinement with human experts (Gottweis et al., 2025). Other systems advance this direction through multi-agent graph reasoning (Ghafarollahi and Buehler, 2025), iterative literature-grounded ideation (Baek et al., 2025), adversarial multi-agent collaboration (Yang et al., 2026), mathematical research assistance (Zheng et al., 2026), and broader research automation pipelines (Liu et al., 2026; Zhang, 2026; Zhang et al., 2026; Zhou et al., 2025). Recent work further situates these systems within the broader shift from AI-for-science tools to autonomous scientific workflows (Ren et al., 2025; Plaat et al., 2025; Zheng et al., 2025a; Islam et al., 2026; Wei et al., 2025; Zhang et al., 2025a,b; Xie et al., 2025). In contrast to work that primarily improves the ability of agents to generate research candidates, we focus

on what happens after generation: how to judge whether the resulting ideas are diverse, novel, and worth further investment.

Evaluating scientific ideas and research agents. Scientific reasoning and research-agent benchmarks increasingly test whether models can operate in realistic scientific settings, including atmospheric science, weather and climate downstream tasks, wireless network intelligence, scientific law discovery, multimodal scientific workflows, and scientific computing or visualization (Li et al., 2025; Shinde et al., 2026; Tong et al., 2026b; Zheng et al., 2025b; Sun et al., 2025; Joseph et al., 2025). At the same time, idea-generation benchmarks and human studies directly examine the creativity of LLM-generated research proposals, often through criteria such as novelty, feasibility, clarity, and impact (Si et al., 2024; Guo et al., 2025; Tong et al., 2026a; Ye et al., 2026). These studies expose the difficulty we target: an idea may be rated as novel while remaining difficult to test, or may appear distinct from prior work only at the level of surface wording. Recent work on agent evaluation and automated peer review likewise emphasizes that judge models require careful calibration and stronger evidence grounding (Zhuge et al., 2024; Zhu et al., 2025; Baumann et al., 2026; Vasu et al., 2025). For early-stage ideas without experiments, code, or a complete manuscript, this motivates proposal-level metrics that capture semantic redundancy, context-conditioned diversity, and literature-level atypicality.

3 Problem Formulation

Let p be a scientific problem statement that specifies the task, data assumptions, target variables, baselines, metrics, and implementation constraints. An idea generator G produces a set of n textual research ideas:

$$X_G(p) = \{x_i\}_{i=1}^{n}, \quad x_i \in \mathcal{X}. \tag{1}$$

Each x_i is an early-stage proposal, typically containing a hypothesis, a method sketch, and planned validation, but not yet a complete experimental artifact. The evaluation object is therefore the set $X_G(p)$, because a useful generator should produce a portfolio of distinct and scientifically plausible directions rather than a single fluent answer.

We define Lens as a set-level evaluator:

$$L(p, X) = \big(h(p, X), c(p, X), \rho(p, X), \zeta(p, X)\big), \tag{2}$$

Algorithm 1 idea generation and evaluation

Require: problem statement p, baseline B, pool size K, budget n
Ensure: Forge ideas X, Lens scores S, relative gains Δ
$M \leftarrow \{h, c, \rho, \zeta\}$
$Y \leftarrow F(p, K)$ ▷ generate a candidate pool with Forge
for each candidate $y_i \in Y$ **do**
  $r_i \leftarrow R(p, y_i)$ ▷ review candidate quality
end for
$T \leftarrow \mathrm{Top}_n(Y, o)$
$X \leftarrow \{U(p, y_i, r_i) : y_i \in T\}$
$S \leftarrow L(p, X)$ ▷ run Lens on Forge ideas
$X_B \leftarrow B(p)$ ▷ generate baseline ideas
$S_B \leftarrow L(p, X_B)$ ▷ run Lens on baseline ideas
$\Delta \leftarrow \{\Delta_m : m \in M\}$ ▷ use Eq. 3
return X, S, Δ

where h and c measure internal diversity, while ρ and ζ measure external literature-level atypicality. The tables report raw metric values: larger is better for h and c, while lower is better for NPMI and atypicality Z-score. For relative comparisons, let $\tilde{m}$ denote the oriented version of metric m, where higher values always indicate better performance. Given a baseline generator B, the relative gain is

$$\Delta_m = \frac{\tilde{m}(p, X_G) - \tilde{m}(p, X_B)}{|\tilde{m}(p, X_B)| + \epsilon}. \tag{3}$$

This formulation turns open-ended ideation into a comparable measurement problem: early-stage ideas have no gold answer or task accuracy, so generators must be judged by set-level signals that can be computed before experiments.

4 Method

IDEAL contains two parts, as illustrated in Figure 2. Forge generates implementation-oriented scientific ideas from raw problem descriptions. Lens evaluates the resulting idea sets using internal and external signals. Algorithm 1 gives the full workflow. The design separates generation from evaluation: Forge may use an LLM reviewer to select candidates, but the final comparison is computed from set-level metrics rather than from a judge model's holistic preference.

4.1 Forge: Idea Generation

Forge is designed to produce comparable, implementation-oriented idea records rather than free-form brainstorming text. It operates directly on the original problem statement p, instead of first compressing it into a short problem card. This matters in scientific settings because data geometry, physical variables, evaluation metrics, and baseline assumptions often determine whether an idea is

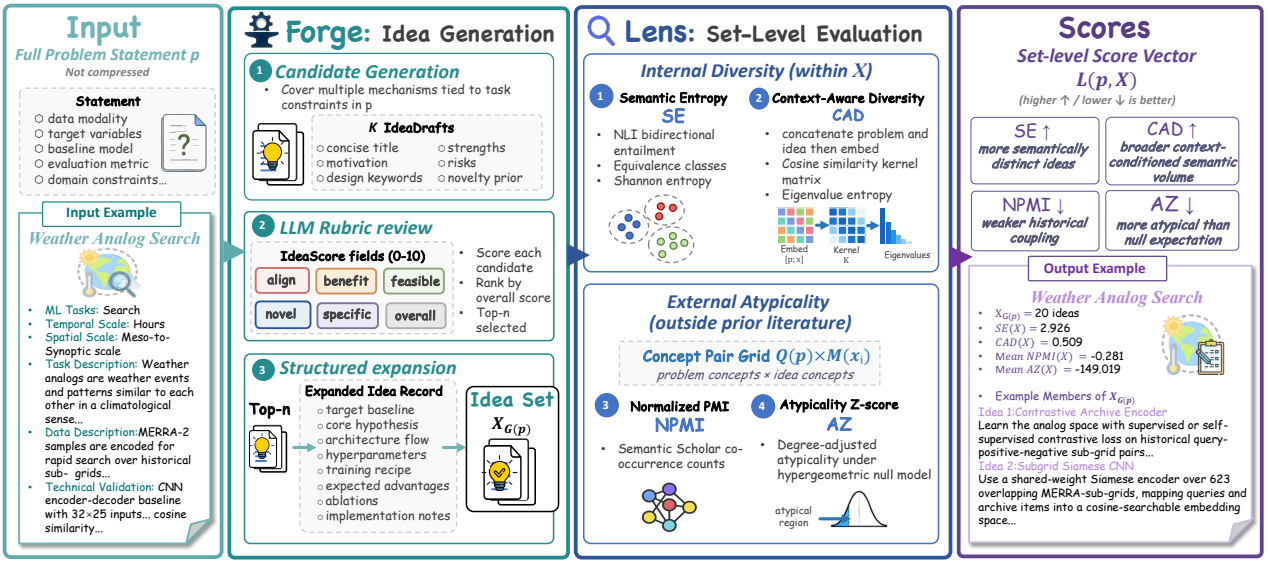

Figure 2: Overview of IDEAL's idea generation and set-level evaluation workflow.

implementable. The first stage is candidate construction: Forge samples a diverse pool

$$Y = F(p, K) = \{y_i\}_{i=1}^{K}, \tag{4}$$

where K is the pool size. Each candidate y_i is a typed draft containing a concise title, motivation, design keywords, expected strengths, risks, and a novelty prior. The purpose of this stage is to cover multiple mechanisms that remain tied to the task constraints in p, rather than to generate many stylistic variants of the same proposal.

The second stage is rubric-based selection. Forge scores each candidate with a strict review function:

$$r_i = R(p, y_i) = \big(a_i, b_i, f_i, v_i, t_i, o_i\big), \tag{5}$$

where a_i, b_i, f_i, v_i, t_i, and o_i denote alignment, expected benefit, feasibility, novelty, specificity, and overall score. This separates proposal search from proposal expansion: candidates that are off-task, vague, or difficult to implement can be filtered before they become full ideas.

The third stage is structured expansion. The highest-ranked candidates are converted into full idea records:

$$X = \{U(p, y_i, r_i) : y_i \in \mathrm{Top}_n(Y, o)\}. \tag{6}$$

The expansion function U standardizes the target baseline, core hypothesis, architecture flow, hyperparameters, training recipe, expected advantage, ablations, and implementation notes. Thus Forge outputs textual proposals with a consistent granularity: they are early-stage enough to be evaluated before experiments, but concrete enough to support later implementation and fair set-level comparison.

4.2 Lens: Idea-Level Evaluation

Lens evaluates an idea set $X = \{x_i\}_{i=1}^{n}$ with two internal metrics and two external metrics. These metrics are chosen to match the two main failure modes of large-scale idea generation. Internally, a generator may appear productive while repeatedly proposing the same direction with different wording; semantic entropy gives a strict redundancy test, while context-aware diversity measures the continuous spread of ideas after conditioning on the original problem. Externally, a fluent idea may still rely on familiar problem-method pairings; NPMI measures how strongly those pairings already co-occur in the literature, while the atypicality score asks whether the observed overlap is lower than expected after accounting for concept popularity. Lens therefore pairs a discrete and a continuous signal for internal diversity, and a raw association and a null-adjusted signal for external novelty.

Semantic entropy. We first group ideas by bidirectional textual entailment, following the use of semantic equivalence classes for uncertainty in generation (Kuhn et al., 2023). Let C_x be the overlapping token chunks of idea x, and let $q(d, c)$ be the entailment probability assigned by an NLI model when chunk d is treated as premise and chunk c as hypothesis. The directional document entailment

from x to y is

$$u(x, y) = \frac{1}{|C_y|} \sum_{c \in C_y} \max_{d \in C_x} q(d, c). \tag{7}$$

Two ideas are treated as semantically equivalent when they entail each other above a threshold τ:

$$e(x_i, x_j) = \mathbf{1} \begin{bmatrix} u(x_i, x_j) \geq \tau \\ \wedge\, u(x_j, x_i) \geq \tau \end{bmatrix}. \tag{8}$$

Let $\mathcal{C}$ be the equivalence classes induced by e. Semantic entropy is the Shannon entropy of the cluster distribution:

$$h(X) = -\sum_{C \in \mathcal{C}} \frac{|C|}{|X|} \log \frac{|C|}{|X|}. \tag{9}$$

A high value means that the generator produces multiple semantically distinct ideas, while a low value indicates collapse into a few repeated directions.

Context-aware diversity. Semantic entropy uses a discrete equivalence relation, so Lens also measures continuous diversity. For each idea, we concatenate the problem and the idea text, embed the resulting context-conditioned text, and normalize the vector:

$$z_i = \frac{\mathrm{Enc}([p; x_i])}{\|\mathrm{Enc}([p; x_i])\|_2}. \tag{10}$$

We then construct a cosine similarity matrix K with entries

$$k(i, j) = z_i^\top z_j. \tag{11}$$

Let λ_i be the eigenvalues of K, and let $\alpha_i = \lambda_i / \sum_{j=1}^{n} \lambda_j$. The normalized context-aware diversity score follows the eigenvalue-entropy view of diversity used by the Vendi score (Friedman and Dieng, 2022):

$$c(p, X) = -\frac{1}{\log n} \sum_{i=1}^{n} \alpha_i \log \alpha_i. \tag{12}$$

This score is high when the ideas span a larger context-conditioned semantic volume.

Literature novelty by NPMI. For external novelty, Lens extracts a small set of problem-side concepts $Q(p)$ and idea-side concepts $M(x_i)$. For a concept pair (q, m), let N be the corpus size and let $N(q)$, $N(m)$, and $N(q, m)$ be Semantic Scholar paper counts. We estimate

$$\begin{aligned} P(q) &= \frac{N(q)}{N}, \\ P(m) &= \frac{N(m)}{N}, \\ P(q, m) &= \frac{N(q, m)}{N}. \end{aligned} \tag{13}$$

The normalized pointwise mutual information (Bouma, 2009) is

$$\mathrm{NPMI}(q, m) = \frac{\log \frac{P(q,m)}{P(q)P(m)}}{-\log P(q, m)}. \tag{14}$$

When $P(q, m) = 0$, we use the lower bound -1. Let $\Omega_i = Q(p) \times M(x_i)$, and write $I(q, m) = \mathrm{NPMI}(q, m)$. Because lower NPMI indicates weaker historical coupling, we orient the idea-level score as

$$\rho(x_i) = -\frac{\sum_{(q,m) \in \Omega_i} w(q, m) I(q, m)}{\sum_{(q,m) \in \Omega_i} w(q, m)}. \tag{15}$$

Atypicality under a null model. NPMI alone can over-penalize popular concepts and understate rare but expected combinations. Motivated by evidence that atypical knowledge combinations can be scientifically valuable (Uzzi et al., 2013), Lens also computes a degree-adjusted atypicality score under a hypergeometric null model. The expected overlap and variance for a pair (q, m) are

$$\begin{aligned} \mu(q, m) &= \frac{N(q)N(m)}{N}, \\ \sigma^2(q, m) &= N(q) \frac{N(m)}{N} \left(1 - \frac{N(m)}{N}\right) \\ &\quad \cdot \frac{N - N(q)}{N - 1}. \end{aligned} \tag{16}$$

Given the observed count $N(q, m)$, the pair-level atypicality is

$$Z(q, m) = \frac{N(q, m) - \mu(q, m)}{\sigma(q, m)}. \tag{17}$$

More negative values indicate that the problem and method concepts co-occur less often than expected under the null. We again orient the idea-level score so that larger is better:

$$\zeta(x_i) = -\frac{\sum_{(q,m) \in \Omega_i} w(q, m) Z(q, m)}{\sum_{(q,m) \in \Omega_i} w(q, m)}. \tag{18}$$

For a generated set, Lens averages the external scores across ideas:

$$\rho(p, X) = \frac{1}{|X|} \sum_{i=1}^{n} \rho(x_i), \qquad \zeta(p, X) = \frac{1}{|X|} \sum_{i=1}^{n} \zeta(x_i). \tag{19}$$

Together, ρ and ζ distinguish ideas that merely sound new from ideas whose problem-method combinations are uncommon in the scientific literature.

Method summary. IDEAL makes early-stage scientific ideation measurable: Forge standardizes how ideas are produced and expanded, while Lens evaluates whether the resulting set is diverse and literature-atypical before experiments are available.

5 Experiments

We evaluate IDEAL on all result sheets in our workbook. The evaluation covers three benchmark groups: six weather and climate tasks from WxC-Bench (Shinde et al., 2026), twenty wireless mathematical reasoning tasks related to WirelessBench-style network intelligence (Tong et al., 2026b), and five cross-domain science tasks, denoted CrossSci, spanning environmental monitoring, remote sensing, and geophysical prediction. For each problem, we compare Forge against five autonomous research baselines: AutoResearchClaw, ARIS, claude-scholar, Idea2Paper, and AI-Scientist-v2. All systems are evaluated under the same idea budget, and Lens is applied to the generated idea sets without using downstream experimental results.

Metrics. We report semantic entropy (SE), context-aware diversity (CAD), normalized pointwise mutual information (NPMI), and atypicality Z-score (AZ). Higher SE and CAD indicate stronger internal diversity. Lower NPMI and AZ indicate weaker historical coupling and more atypical problem-method combinations. For aggregate comparisons, we use the oriented relative improvement in Eq. 3, so improvements in NPMI and AZ are computed after reversing their direction.

Baselines. The baselines cover both end-to-end AutoResearch systems and agentic ideation frameworks. AutoResearchClaw represents a fully automated idea-to-paper pipeline (Liu et al., 2026), ARIS uses adversarial multi-agent collaboration (Yang et al., 2026), claude-scholar is a semi-automated research assistant (Zhang, 2026), Idea2Paper is a research generation baseline, and AI-Scientist-v2 is the strongest non-ours system on several idea-level metrics (Yamada et al., 2025). Because several baselines are configurable harnesses rather than single fixed checkpoints, we describe documented default or evaluation backends here rather than treating each framework as one immutable model. AutoResearchClaw documents GPT-4o as the primary model with GPT-4o-mini fallback, ARIS uses a cross-family executor-reviewer setup, claude-scholar is primarily a Claude Code workflow while supporting Codex CLI and OpenCode, AI-Scientist-v2 reports GPT-4o and Claude 3.5 Sonnet in its model configurations, and public Idea2Paper-style artifacts do not expose a single stable checkpoint, though related Idea2Story studies report GLM-4.7 in qualitative comparisons. This set lets us compare against systems that differ in search strategy, agent structure, model routing, and degree of automation.

Framework	SE ↑	CAD ↑	NPMI ↓	AZ ↓
AI-Scientist-v2	1.878	0.284	0.259	800.519
Idea2Paper	0.417	0.246	0.257	535.090
AutoResearchClaw	1.156	0.225	0.355	1114.821
ARIS	0.942	0.198	0.393	786.690
claude-scholar	1.101	0.155	0.315	721.588
Ours	**2.769**	**0.431**	**-0.080**	**6.951**

Table 1: Main Lens results averaged over 31 tasks. Bold marks the best value in each metric.

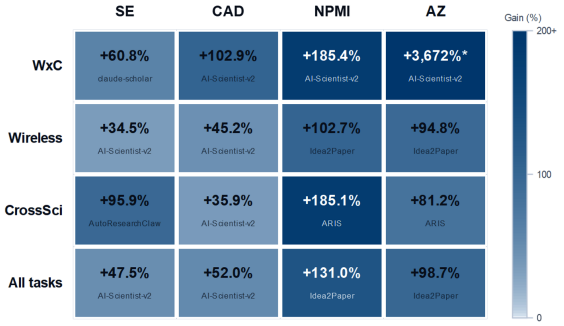

Figure 3: Relative Lens gains over the strongest non-ours baseline by benchmark group and metric. Color intensity is capped at 200%, and starred cells exceed the cap.

Main results. Table 1 gives the main comparison. Across 31 tasks, IDEAL has the best aggregate mean on all four Lens metrics, with average gains of 47.5% on SE, 52.0% on CAD, 131.0% on

Framework	WxC-Bench				Wireless Math Bench				CrossSci			
	SE	CAD	NPMI	AZ	SE	CAD	NPMI	AZ	SE	CAD	NPMI	AZ
AI-Scientist-v2	1.786	0.204	-0.071	3.376	2.108	0.318	0.339	565.978	1.069	0.243	0.331	2695.251
AutoResearchClaw	1.357	0.079	0.276	944.352	1.098	0.295	0.366	839.945	1.147	0.123	0.408	2418.887
ARIS	0.598	0.054	0.269	818.873	1.054	0.273	0.465	863.511	0.904	0.072	0.255	440.790
claude-scholar	1.856	0.064	-0.009	122.187	0.933	0.214	0.422	959.746	0.868	0.028	0.277	488.232
Idea2Paper	0.099	0.147	-0.019	558.068	0.483	0.308	0.329	503.153	0.536	0.115	0.301	635.264
Ours	**2.984**	**0.413**	**-0.201**	**-120.601**	**2.835**	**0.462**	**-0.009**	**26.249**	**2.247**	**0.330**	**-0.217**	**82.823**

Table 2: Average Lens scores for every framework and benchmark group. Best values in each column are bold.

oriented NPMI, and 98.7% on oriented AZ over the strongest non-ours baseline for each metric. The strongest non-ours baseline differs by metric: AI-Scientist-v2 is closest on internal diversity, while Idea2Paper is closest on the external literature-level metrics. Figure 3 visualizes these oriented gains by benchmark group and metric. This breadth is important because the benchmark groups test different failure modes. WxC tasks require ideas to respect geophysical variables and spatial structure, wireless tasks impose mathematical and communication-theoretic constraints, and CrossSci mixes environmental and remote-sensing settings. The same trend across these regimes suggests that the gain is not tied to a single domain template.

The metric split clarifies what changes in the generated sets. SE and CAD indicate that Forge produces more distinct directions, rather than many close paraphrases of the same proposal. The larger movement is on the external metrics: lower NPMI and AZ mean that the ideas connect problem concepts with method concepts that are less conventional in the literature. Taken together, the scores point to a useful pre-experimental signal: IDEAL expands the set of candidate directions and moves that set away from familiar literature pairings. This does not imply that every idea will work empirically, but it gives a measurable way to separate broad ideation from fluent repetition before training runs or full manuscripts exist.

Table 2 breaks the aggregate result down by domain. The wireless benchmark is especially useful as a stress test, since its problems have explicit mathematical structure; even there, IDEAL improves CAD by 45.2% over AI-Scientist-v2 and lowers AZ from the strongest non-ours average of 503.153 to 26.249. On WxC-Bench, the near-maximum SE score indicates that the generated ideas occupy almost all available equivalence classes, rather than collapsing into repeated weather-model variants. CrossSci shows the same pattern in a less homogeneous collection of scientific tasks.

The baselines reveal why the four Lens metrics should be read together. AI-Scientist-v2 is a strong non-ours baseline on SE and CAD, so it can produce idea sets that look diverse internally, but its external novelty scores remain closer to familiar literature combinations. Idea2Paper is more competitive on some external novelty comparisons, especially in wireless tasks, but its low SE points to a narrower set of directions. Lens makes this tradeoff visible: a useful idea generator should not only spread proposals apart from one another, but also move them beyond the most expected problem-method pairings.

Variant	SE ↑	CAD ↑	NPMI ↓	AZ ↓
w/o raw problem	0.974	0.096	0.174	527.719
w/o reviewer	0.679	0.134	0.109	355.815
w/o expansion	0.651	0.060	0.146	429.723
w/o structure	0.409	0.038	0.143	519.590
novelty-only	0.824	0.154	0.068	182.790
feasibility-only	0.893	0.119	0.073	201.504
Full IDEAL	**1.378**	**0.268**	**-0.198**	**-132.884**

Table 3: Ablation results averaged over the six WxC-Bench tasks. Each run selects five final ideas from ten candidates.

Ablation study. Table 3 isolates the main design choices in Forge on WxC-Bench. Using the raw problem statement, reviewer reranking, structured fields, and expansion all improve the final idea set, with the largest drops appearing when structure or expansion is removed. Single-objective reranking by novelty or feasibility also underperforms the full rubric, suggesting that useful scientific ideas require balancing originality with task fit and implementability.

Framework	Absolute judge					Pairwise judge				
	Clar.	Nov.	Feas.	Impact	Overall	Logic	Feas.	Robust	Impact	Overall
AI-Scientist-v2	8.17	**7.78**	7.17	**8.55**	**7.92**	4.81	4.56	3.63	**5.48**	4.62
AutoResearchClaw	7.06	6.66	6.88	7.04	6.91	4.20	4.18	3.24	4.68	4.08
ARIS	7.60	6.18	7.28	7.04	7.03	4.48	4.56	3.32	4.94	4.33
claude-scholar	**8.60**	5.94	**8.34**	7.80	7.67	**4.84**	4.60	**3.68**	5.22	4.59
Idea2Paper	5.51	7.20	5.33	7.14	6.29	4.37	3.89	3.23	5.02	4.13
Ours	8.08	6.56	7.38	7.96	7.50	4.70	**4.84**	3.56	**5.48**	**4.64**

Table 4: LLM-as-a-Judge results on the CrossSci tasks. Absolute judging scores ideas on clarity, novelty, feasibility, impact, and overall quality. Pairwise judging compares logic, feasibility, robustness, impact, and overall preference.

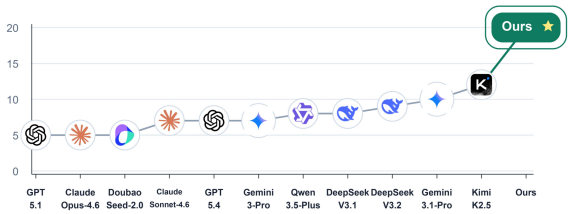

Figure 4: Wireless verification results over 20 cases. Points show solved-case counts for direct model runs ordered by performance; Ours denotes IDEAL with DeepSeek-V3.2 as the backbone.

Agreement metric	Value
Accuracy	77.33%
Cohen's κ	0.6139
Mean Spearman ρ	0.8085

Table 5: Agreement between human experts and the LLM pairwise judge.

Wireless verification. Figure 4 reports an additional wireless validation set with 20 cases. Forge solves all 20 cases with DeepSeek-V3.2 as the backbone, while the strongest standalone model reaches 12 correct cases. This result suggests that the gains in Lens are not only artifacts of textual scoring, the pipeline also improves the production of correct technical outputs.

LLM-as-a-Judge analysis. Table 4 provides a complementary judge-based view on CrossSci. The full Forge judge protocol remains inexpensive in absolute terms, costing \$0.617 per CrossSci task on average across the absolute-judge and pairwise-judge settings. The absolute judge favors AI-Scientist-v2 overall, while the pairwise judge is closer: IDEAL has the best overall score, the best feasibility score, and ties for impact. This contrast supports our motivation for Lens, since holistic judge scores can mix rhetorical quality, perceived novelty, and feasibility rather than directly measuring set-level diversity and literature atypicality.

Human–LLM agreement. Table 5 checks whether the pairwise LLM judge is aligned with human preference rankings. We use five generated scientific problems, each with ten candidate ideas, yielding 225 within-problem idea pairs. To reduce position bias, every pair is judged in both orders; inconsistent swapped decisions are treated as ties. Two human experts provide the reference judgments, with any expert disagreement also downgraded to a tie. The resulting agreement is substantial: the LLM judge matches the human consensus in 77.33% of pairwise decisions, reaches a Cohen's κ of 0.6139, and obtains a mean Spearman ρ of 0.8085 over per-problem idea rankings.

Experimental summary. It cautions against treating recent end-to-end AutoResearch systems as evidence that creative scientific discovery can already be produced reliably at scale; progress on the full pipeline should not obscure the need to study and measure each component in depth. Without component-level measurement, an apparently successful end-to-end run can hide whether the limiting factor is idea search, proposal selection, implementation, or writing quality.

6 Conclusion

We introduced IDEAL, an idea-level framework for generating and evaluating early-stage scientific proposals. Forge produces implementation-oriented ideas directly from problem statements, while Lens measures whether the resulting idea sets are semantically diverse and atypical with respect to the scientific literature. Across cross-domain science tasks, the results show that idea generation should be evaluated as a set-level process rather than as isolated fluent proposals.

Limitations

IDEAL evaluates early-stage scientific ideas before experiments are run, so its scores should not be interpreted as evidence that an idea is correct, feasible, or scientifically valuable. Lens measures semantic diversity and literature-level atypicality, but it does not replace domain expertise, empirical validation, or peer review. The external novelty metrics also depend on concept extraction and literature-count coverage, which may vary across domains and search infrastructure.

Ethical Considerations

IDEAL is intended to support human researchers by making large pools of generated ideas easier to inspect. It should not be used to automatically approve, deploy, or prioritize scientific claims without expert review. This is especially important in domains where proposed research may affect public safety, health, infrastructure, or environmental policy. We report generation and evaluation details to make the framework auditable and to reduce the risk of treating automatic novelty signals as substitutes for scientific judgment.

References

Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. 2025. Researchagent: Iterative research idea generation over scientific literature with large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6709–6738.

Joachim Baumann, Jiaxin Pei, Sanmi Koyejo, and Dirk Hovy. 2026. Stop automating peer review without rigorous evaluation. *arXiv preprint arXiv:2605.03202*.

Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, 30:31–40.

Dan Friedman and Adji Bousso Dieng. 2022. The vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2210.02410*.

Alireza Ghafarollahi and Markus J Buehler. 2025. Sciagents: automating scientific discovery through bioinspired multi-agent intelligent graph reasoning. *Advanced Materials*, 37(22):2413523.

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

Sikun Guo, Amir Hassan Shariatmadari, Guangzhi Xiong, Albert Huang, Myles Kim, Corey M Williams, Stefan Bekiranov, and Aidong Zhang. 2025. 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.

Md Asadul Islam, Subbulakshmi Somu, and Faraj Mazyed Faraj Aldaihani. 2026. The rise of agentic ai: Synthesis of current knowledge and future research agenda. *Global Business and Organizational Excellence*, 45(3):402–416.

Sebastian Joseph, Syed M Husain, Stella Offner, Stéphanie Juneau, Paul Torrey, Adam Bolton, Juan Farias, Niall Gaffney, Greg Durrett, and Junyi Jessy Li. 2025. Astrovisbench: A code benchmark for scientific computing and visualization in astronomy. *Advances in Neural Information Processing Systems*, 38.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Chenyue Li, Wen Deng, Mengqian Lu, and Binhang Yuan. 2025. Atmossci-bench: evaluating the recent advance of large language model for atmospheric science. *Advances in Neural Information Processing Systems*, 38.

Jiaqi Liu, Peng Xia, Siwei Han, Shi Qiu, Letian Zhang, Guiming Chen, Haoqin Tu, Xinyu Yang, Jiawei Zhou, Hongtu Zhu, Yun Li, Jiaheng Zhang, Yuyin Zhou, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. 2026. Autoresearchclaw: Fully autonomous research from idea to paper.

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

Aske Plaat, Max van Duijn, Niki Van Stein, Mike Preuss, Peter van der Putten, and Kees Joost Batenburg. 2025. Agentic large language models, a survey. *Journal of Artificial Intelligence Research*, 84.

Shuo Ren, Can Xie, Pu Jian, Zhenjiang Ren, Chunlin Leng, and Jiajun Zhang. 2025. Towards scientific intelligence: A survey of llm-based scientific agents. *arXiv preprint arXiv:2503.24047*.

Rajat Shinde, Kumar Ankur, Christopher E Phillips, Aman Gupta, Simon Pfreundschuh, Sujit Roy, Sheyenne Kirkland, Vishal Gaur, Venkatesh Kolluru, Amy Lin, and 1 others. 2026. Wxc-bench: A novel dataset for weather and climate downstream tasks. *Scientific Data*, 13(1):596.

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

Qiushi Sun, Zhoumianze Liu, Chang Ma, Zichen Ding, Fangzhi Xu, Zhangyue Yin, Haiteng Zhao, Zhenyu Wu, Kanzhi Cheng, Zhaoyang Liu, and 1 others. 2025. Scienceboard: Evaluating multimodal autonomous agents in realistic scientific workflows. *arXiv preprint arXiv:2505.19897*.

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

Jingwen Tong, Fang Liu, Linkai Xv, Shiliang Lu, Kangqi Li, Yiqian Zhang, Yijie Song, Zeyang Xue, and Jun Zhang. 2026b. Wirelessbench: A tolerance-aware llm agent benchmark for wireless network intelligence. *arXiv preprint arXiv:2603.21251*.

Brian Uzzi, Satyam Mukherjee, Michael Stringer, and Ben Jones. 2013. Atypical combinations and scientific impact. *Science*, 342(6157):468–472.

Rosni Vasu, Peter Jansen, Pao Siangliulue, Cristina Sarasua, Abraham Bernstein, Peter Clark, and Bhavana Dalvi Mishra. 2025. Harpa: A testability-driven, literature-grounded framework for research ideation. *arXiv preprint arXiv:2510.00620*.

Jiaqi Wei, Yuejin Yang, Xiang Zhang, Yuhan Chen, Xiang Zhuang, Zhangyang Gao, Dongzhan Zhou, Guangshuai Wang, Zhiqiang Gao, Juntai Cao, and 1 others. 2025. From ai for science to agentic science: A survey on autonomous scientific discovery. *arXiv preprint arXiv:2508.14111*.

Qiujie Xie, Yixuan Weng, Minjun Zhu, Fuchen Shen, Shulin Huang, Zhen Lin, Jiahui Zhou, Zilan Mao, Zijie Yang, Linyi Yang, and 1 others. 2025. How far are ai scientists from changing the world? *arXiv preprint arXiv:2507.23276*.

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

Ruofeng Yang, Yongcan Li, and Shuai Li. 2026. Aris: Autonomous research via adversarial multi-agent collaboration. *arXiv preprint arXiv:2605.03042*.

Haotian Ye, Haowei Lin, Jingyi Tang, Yizhen Luo, Caiyin Yang, Chang Su, Rahul Thapa, Rui Yang, Ruihua Liu, Zeyu Li, and 1 others. 2026. Evaluation-driven scaling for scientific discovery. *arXiv preprint arXiv:2604.19341*.

Gaorui Zhang. 2026. Claude scholar: Semi-automated research assistant for academic research and software development. https://github.com/Galaxy-Dawn/claude-scholar. GitHub repository.

Haoxuan Zhang, Ruochi Li, Yang Zhang, Ting Xiao, Jiangping Chen, Junhua Ding, and Haihua Chen. 2025a. The evolving role of large language models in scientific innovation: Evaluator, collaborator, and scientist. *arXiv preprint arXiv:2507.11810*.

Jiayi Zhang, Yongfeng Gu, Jianhao Ruan, Maojia Song, Yiran Peng, Zhiguang Han, Jinyu Xiang, Zhitao Wang, Caiyin Yang, Yixi Ouyang, and 1 others. 2026. Harnessing agentic evolution. *arXiv preprint arXiv:2605.13821*.

Wenlin Zhang, Xiaopeng Li, Yingyi Zhang, Pengyue Jia, Yichao Wang, Huifeng Guo, Yong Liu, and Xiangyu Zhao. 2025b. Deep research: A survey of autonomous research agents. *arXiv preprint arXiv:2508.12752*.

Daniel Zheng, Ingrid von Glehn, Yori Zwols, Iuliya Beloshapka, Lars Buesing, Daniel M Roy, Martin Wattenberg, Bogdan Georgiev, Tatiana Schmidt, Andrew Cowie, and 1 others. 2026. Ai co-mathematician: Accelerating mathematicians with agentic ai. *arXiv preprint arXiv:2605.06651*.

Tianshi Zheng, Zheye Deng, Hong Ting Tsang, Weiqi Wang, Jiaxin Bai, Zihao Wang, and Yangqiu Song. 2025a. From automation to autonomy: A survey on large language models in scientific discovery. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17744–17761.

Tianshi Zheng, Kelvin Kiu-Wai Tam, Newt Hue-Nam K Nguyen, Baixuan Xu, Zhaowei Wang, Jiayang Cheng, Hong Ting Tsang, Weiqi Wang, Jiaxin Bai, Tianqing Fang, and 1 others. 2025b. Newtonbench: Benchmarking generalizable scientific law discovery in llm agents. *arXiv preprint arXiv:2510.07172*.

Lianhao Zhou, Hongyi Ling, Cong Fu, Yepeng Huang, Michael Sun, Wendi Yu, Xiaoxuan Wang, Xiner Li, Xingyu Su, Junkai Zhang, and 1 others. 2025. Autonomous agents for scientific discovery: Orchestrating scientists, language, code, and physics. *arXiv preprint arXiv:2510.09901*.

Minjun Zhu, Yixuan Weng, Linyi Yang, and Yue Zhang. 2025. Deepreview: Improving llm-based paper review with human-like deep thinking process. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29330–29355.

Mingchen Zhuge, Changsheng Zhao, Dylan Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, and 1 others. 2024. Agent-as-a-judge: Evaluate agents with agents. *arXiv preprint arXiv:2410.10934*.

A Experimental Protocol

This appendix provides the experimental details behind the main evaluation: benchmark membership, aggregation rules, generation settings, prompt fields, and implementation choices for Lens.

A.1 Benchmark composition

Table 6 lists the three benchmark groups used in the main evaluation. Each input is a full problem statement, not a short keyword prompt. When the source provides them, the statement includes the data modality, target variables, baseline model, evaluation metric, and domain constraints. For a given problem, every system receives the same statement. Table 7, the WxC case cards in Section A.1.1, and Table 8 give the task-level details used to interpret the benchmark groups.

The groups differ on purpose. WxC-Bench stresses gridded physical variables and spatial-temporal structure. Wireless Math stresses symbolic consistency. CrossSci contains five less standardized scientific tasks, so it is useful for checking whether a method overfits to one benchmark style.

A.1.1 WxC-Bench cases

The six WxC-Bench tasks are listed below as short case cards. We use cards rather than a wide table here because the important information is not a single number; it is the kind of structure an idea generator must preserve.

Aviation Turbulence Prediction

Problem. Predict aviation turbulence from atmospheric fields and flight-level context. The task involves sparse event labels and vertically structured meteorological predictors.
What matters. A useful idea must handle pressure-level structure, spatial context, class imbalance, and altitude-specific prediction. A flat tabular classifier would miss much of the meteorological geometry.

Generation of Natural Language-based Weather Forecast Reports

Problem. Generate natural-language forecast discussions from weather inputs such as maps, gridded fields, or forecast products. The output must be readable text, but it also has to stay grounded in the weather state.
What matters. The idea should connect weather representations to text generation while controlling factual consistency, regional detail, and forecast-style phrasing.

Gravity Wave (GW) Parameterization

Problem. Estimate gravity-wave effects or momentum fluxes from atmospheric state variables. The task depends on vertical coupling, global geometry, and physically meaningful target quantities.
What matters. The idea should respect latitude-longitude geometry, vertical levels, conservation-oriented constraints, and the difference between local column information and global wave structure.

Hurricane Prediction and Intensity Estimation

Problem. Predict hurricane-related quantities such as track, intensity, wind speed, or pressure from weather fields and storm context.
What matters. The idea should preserve storm-centered structure, temporal evolution, basin context, and the coupling between broad atmospheric fields and localized cyclone dynamics.

Long-range Precipitation Forecasting

Problem. Forecast precipitation at extended lead times using atmospheric, oceanic, or satellite-derived signals. The problem mixes long temporal dependencies with noisy spatial fields.
What matters. The idea should handle lead-time conditioning, multi-scale spatial features, uncertainty, and the weak signal-to-noise ratio typical of subseasonal precipitation prediction.

Group	Cases	Source	Evaluation role
WxC-Bench	6	WxC-Bench (Shinde et al., 2026)	Weather and climate tasks with spatial-temporal variables, gridded physical fields, and domain constraints. This group probes whether generated ideas preserve data geometry and physical assumptions.
Wireless Math	20	Wireless reasoning tasks related to WirelessBench-style evaluation (Tong et al., 2026b)	Communication-system problems with equation-heavy structure. This group probes whether ideas remain mathematically grounded rather than merely plausible at the level of prose.
CrossSci	5	Curated cross-domain scientific problems	Heterogeneous scientific tasks outside a single benchmark family. This group probes transfer to less standardized scientific problem statements; the individual tasks are listed in Table 8.

Table 6: Benchmark composition. The main Lens evaluation contains 31 problems and 124 task-metric cases.

Wireless task family	Included problems	Main technical content
SIM and CSIM channel models	CSIM Channel Covariance Model; CSIM Effective Channel Model; SIM Layer Phase-Shift Matrix; SIM Multiuser Sum Rate; SIM User Received Signal.	These cases ask whether an idea can preserve the layered surface model, phase-shift operations, effective channels, and sum-rate objectives instead of treating the problem as generic sequence modeling.
Least-squares channel estimation	LS Channel Denoising Model; LS Channel Estimator with Pseudoinverse; LS Channel-Noise Relation; Noisy LS Denoising Equation; Single-Reflection LS Estimator.	These cases stress linear inverse structure, noise terms, pseudoinverse operations, and the relation between measurement equations and estimator design.
MMSE estimation	MMSE Channel Estimate with Pilot Noise; MMSE Estimation Quality Equation; MMSE Path-Loss Channel Estimator; MMSE Pilot Observation Estimator; MMSE Pilot-Based Channel Estimation; MMSE User Channel Estimation.	These cases require ideas to track covariance, pilot observations, path-loss terms, and the difference between LS-style and MMSE-style estimators.
Array and propagation geometry	Planar Array LoS Channel; Rician LoS Vector Expression; Satellite LoS Channel Vector.	These cases test whether an idea keeps antenna geometry, line-of-sight vectors, Rician structure, and satellite-user geometry intact.
Tensor and unfolding identities	Tensor Mode-2 Khatri-Rao Unfolding.	This case focuses on tensor algebra, unfolding conventions, and Khatri-Rao structure, which are easy to corrupt if the generator only paraphrases the equation.

Table 7: Wireless Math task families. The workbook contains 20 wireless problems; they are grouped here by the mathematical structure most relevant to idea generation.

Weather Analog Search

Problem. Retrieve historically similar weather situations for a given query state. This is naturally a metric-learning or retrieval task rather than ordinary classification.
What matters. The idea should define a searchable embedding space that preserves meteorological similarity, variable weighting, spatial alignment, and efficient nearest-neighbor retrieval.

A.2 Evaluation unit and aggregation

Lens evaluates a set of ideas X for one problem p. For each generator-problem pair (G, p), we compute four raw scores: semantic entropy, context-aware diversity, NPMI, and atypicality Z-score. We then average each metric over problems within a benchmark group.

Table 1 reports framework-level averages over the 31 problems and metric-level comparisons against the strongest non-ours baseline. One problem contributes four Lens measurements, one per metric, so the full workbook contains 124 task-metric measurements, but the main table summarizes performance through aggregate metric values and strongest-baseline gaps. SE and CAD are higher-is-better. NPMI and AZ are lower-is-better in their raw form. For relative gains, we orient NPMI and AZ by reversing their sign before applying Eq. 3. This is why the tables can report raw NPMI/AZ with downward arrows while still reporting positive improvement percentages.

A.3 Generation protocol

Forge uses three LLM calls. It first samples candidate drafts from the original problem statement, then scores those drafts with a fixed review rubric, and finally expands selected drafts into implementation-oriented proposals. The original problem statement is included in all three calls. We do not insert an LLM-written problem summary between the input and the generator, because summaries can drop details that matter for scientific tasks.

For the Lens experiments, each evaluated problem is represented as a fixed-size idea set of 20 idea records per system. This fixed set size matters

CrossSci task	Scientific setting	Evaluation focus
Earthquake Aftershock Forecasting	Geophysical hazard forecasting from sparse and noisy seismic observations.	Event-history structure, spatial fault context, and uncertainty-aware prediction under sparse observations.
Microplastic Source Inference	Environmental source apportionment from indirect observations of microplastic transport.	Latent-source inference from indirect contamination patterns under transport and sampling uncertainty.
Sea-Ice Lead Detection	Polar remote sensing for detecting linear openings in sea ice.	Detection of thin spatial structures under sensor noise and severe class imbalance.
Seagrass Health from Remote Sensing	Coastal ecosystem monitoring from multi-spectral or spatial observations.	Linking remote-sensing signals to ecological condition while accounting for water, substrate, and seasonal confounders.
Soil-Moisture Pulse Prediction	Hydrological prediction of transient moisture responses after rainfall or irrigation.	Modeling short-lived hydrological pulses, lagged drivers, and local soil-vegetation controls.

Table 8: CrossSci task details.

Stage	Temperature	Structured record
Candidate generation	0.50	`IdeaDraft`
Candidate review	0.20	`IdeaScore`
Idea expansion	0.35	`ExpandedIdea`
Format repair	0.10	`ExpandedIdea`

Table 9: Generation settings used by Forge. The repair pass is invoked only when the expansion stage fails to produce a parseable structured record.

for SE and CAD, since both are set-level diversity metrics. The public Forge code exposes the number of emitted ideas as a command-line option; the qualitative Markdown reports in the repository are examples, while the workbook results use the fixed evaluation budget.

A.4 Pipeline implementation notes

The Forge command-line entry point accepts either one Markdown file or a directory of Markdown files, writes one output Markdown file per input, and preserves the input filename. Internally, the run follows a direct pipeline: parse the raw problem, generate candidate drafts, review and sort them, expand the selected candidates, and render a report.

Raw-problem design

Forge does not ask the model to first write a compact problem card. The full Markdown statement is passed to candidate generation, review, and expansion. This choice is not cosmetic. In the benchmark tasks, details such as pressure levels, lead times, evaluation metrics, physical variables, and baseline assumptions often decide whether an idea is technically meaningful.

Structured decoding

Each LLM call requests JSON and is decoded into a typed record: `IdeaDraft`, `IdeaScore`, or `ExpandedIdea`. The decoder checks required strings, numeric scores, lists, and hyperparameter name-value pairs before rendering. This makes failed or malformed generations visible instead of silently mixing them into the final report.

Endpoint compatibility

The client uses an OpenAI-compatible chat-completions endpoint and asks for `response_format={type: json_object}` when available. If the endpoint rejects that option, the request is retried without it. The implementation also handles gzip responses and raises explicit errors for empty or invalid endpoint responses.

Conservative review fallback

If the reviewer returns scores for some candidates but omits one candidate ID, Forge assigns that candidate a conservative fallback score and marks the review coverage as incomplete. This keeps the run inspectable while avoiding an accidental advantage from missing review data.

A.5 Prompt templates

The prompts below show the fields requested at each stage and the input blocks inserted by the pipeline. In each run, `[full problem Markdown]` is replaced by the original benchmark problem statement without an intermediate summary.

Candidate generation

Candidate Generation Prompt

System. You are a senior scientific research architect. Given a scientific problem description, propose diverse candidate model architectures. Return a structured response with a top-level `candidates` array. Each candidate must be specific, implementation-oriented, and tailored to the problem. Do not use placeholders; provide concrete defaults.

User. Generate k candidate architecture ideas for the following problem Markdown. The candidates should differ in mechanism, not just in wording. Work directly from the original Markdown. Preserve datasets, targets, baselines, evaluation metrics, constraints, and physical assumptions. Return fields: `idea_id`, `title`, `one_liner`, `motivation`, `design_keywords`, `novelty_prior`, `expected_strengths`, and `risks`.

Inserted input. `[full problem Markdown]`

Candidate review

Candidate Review Prompt

System. You are a strict scientific proposal reviewer. Score each candidate for alignment, uplift potential, feasibility, novelty, and specificity. Return a structured response with a top-level `reviews` array. Use scores from 0 to 10 and be critical but fair.

User. Review the candidate architectures for the original problem Markdown. Return fields: `idea_id`, `alignment`, `uplift_potential`, `feasibility`, `novelty`, `specificity`, `overall`, `rationale`, and `risk_flags`. Use the original Markdown as the authority when judging task fit.

Inserted inputs. `[full problem Markdown]` and `[candidate drafts as JSON]`.

Idea expansion

Idea Expansion Prompt

System. You are a senior scientific systems architect. Expand the selected candidate into a concrete model design. Return a structured response. Every architecture step must be specific enough to implement.

User. Expand this candidate into a detailed architecture proposal. Use the original Markdown as the authoritative problem description. Respect the input/output structure. Provide tensor handling, encoder or decoder choices, output heads, losses, optimizer settings, training defaults, and realistic ablations. Do not assume a fixed scientific domain. Return fields: `target_baseline`, `hypothesis`, `architecture_flow`, `hyperparameters`, `training_recipe`, `why_better`, `ablations`, and `implementation_notes`.

Inserted inputs. `[full problem Markdown]`, `[selected candidate draft]`, and `[review score record]`.

A.6 Structured output fields

Table 10 lists the records used by Forge. Keeping these fields separate makes it easier to inspect a run after generation: motivations, review scores, risks, hypotheses, ablations, and implementation notes are not collapsed into one free-form paragraph.

A.7 Qualitative examples

The examples below are shortened from Forge outputs. They are included to show the level of specificity in the generated records. Lens is computed on the full idea sets, not on these excerpts.

Aviation Turbulence Prediction: 3D Pressure-Slab CNN

The idea treats each day as a 3D atmospheric slab over pressure levels and uses a shared residual CNN backbone with altitude-specific heads for low, middle, and high flight levels. This example checks whether Forge preserves gridded meteorological structure instead of turning the task into tabular prediction.

Weather Analog Search: Multi-Parameter Siamese Encoder

The idea embeds query and archive sub-grids into a cosine-searchable retrieval space using shared-weight CNN towers, multi-channel weather variables, and location conditioning. This example shows how a retrieval task becomes an architecture with indexing and contrastive training details.

Gravity Wave Parameterization: Spherical 3D U-Net

The idea predicts zonal and meridional gravity-wave momentum fluxes with a globe-aware 3D encoder-decoder, longitude-periodic handling, latitude weighting, and attention-gated skip connections. This example illustrates how physical geometry enters the model design.

Object	Fields
IdeaDraft	idea_id, title, one_liner, motivation, design_keywords, novelty_prior, expected_strengths, risks.
IdeaScore	alignment, uplift_potential, feasibility, novelty, specificity, overall, rationale, risk_flags.
ExpandedIdea	target_baseline, hypothesis, architecture_flow, hyperparameters, training_recipe, why_better, ablations, implementation_notes.

Table 10: Structured records used by Forge.

Hurricane Prediction and Intensity Estimation: FourCastNet+Track Head

The idea adds storm-center, track, and intensity heads to a Fourier-ViT weather backbone so that field prediction and hurricane-specific quantities are trained jointly. This example shows how a known scientific backbone can be adapted with task-specific heads and losses.

A.8 Lens implementation notes

Semantic entropy uses `facebook/bart-large-mnli` as the NLI model. Long ideas are split into overlapping token chunks. To score $x \to y$, each chunk of y is matched to the best-entailing chunk of x, and the best-match probabilities are averaged. Two ideas are merged only when the entailment test passes in both directions. The final score is Shannon entropy over the induced equivalence classes.

Context-aware diversity uses `sentence-transformers/all-MiniLM-L6-v2`. For each idea, the original problem and idea text are concatenated before embedding. Long texts are chunked, chunk embeddings are combined with a token-weighted mean, and the resulting vectors are L2-normalized. CAD is the entropy of the normalized eigenvalues of the cosine similarity kernel.

For NPMI and AZ, we use controlled concept lists for the problem side and idea side. Paper counts come from the Semantic Scholar search endpoint and are cached. NPMI is reported as a raw association score, where lower values indicate weaker historical coupling. AZ compares observed co-occurrence with a hypergeometric null; lower, especially negative, values indicate more atypical combinations after accounting for concept popularity.

We report the four Lens metrics separately. A single scalar would hide useful failure modes: a system can be diverse but conventional, or unusual on one idea while repetitive across the set.

A.9 Metric sanity checks

The four Lens metrics are not meant to be interchangeable. The short checks below state what each metric can and cannot say.

Semantic entropy bounds. If all n ideas fall into one equivalence class, then $h(X) = 0$. If every idea forms its own class, then $p_i = 1/n$ for all i, so

$$h(X) = -\sum_{i=1}^{n} \frac{1}{n} \log \frac{1}{n} = \log n.$$

Thus semantic entropy is bounded by $0 \leq h(X) \leq \log n$. This is why the idea budget must be fixed when comparing systems.

Context-aware diversity bounds. CAD normalizes the entropy of the eigenvalue distribution by $\log n$. If all context-conditioned idea embeddings collapse to the same direction, the kernel is close to rank one and the normalized entropy is near zero. If the ideas spread across independent directions, the eigenvalues are more uniform and the score approaches one.

Why NPMI and AZ are reported separately. NPMI measures association strength after normalization by co-occurrence probability. It is useful for detecting familiar problem-method pairs, but rare counts can be noisy. AZ instead compares the observed overlap with a hypergeometric null that preserves the marginal popularity of each concept. In practice, NPMI is the raw association signal, while AZ is the null-adjusted atypicality signal.

A.10 Judge and verification protocols

The LLM-as-a-Judge analysis is a secondary diagnostic. Absolute judging scores ideas for clarity, novelty, feasibility, impact, and overall quality. Pairwise judging compares systems on logic, feasibility, robustness, impact, and overall preference. We keep these results separate from Lens because judge scores can reward fluent writing even when the idea set is repetitive.

The wireless verification experiment uses 20 structured wireless cases. Forge is paired with

Metric	Main resource	Aggregation	Reported direction
Semantic entropy	`facebook/bart-large-mnli` NLI model	Bidirectional chunk entailment, union-find equivalence classes, then Shannon entropy over cluster sizes.	Higher is better; high values mean the idea set contains fewer semantic duplicates.
Context-aware diversity	`all-MiniLM-L6-v2` sentence embedding model	Embed problem-plus-idea texts, build a cosine kernel, normalize its eigenvalues, and compute eigenvalue entropy.	Higher is better; high values mean broader spread after conditioning on the problem.
NPMI	Semantic Scholar paper-count endpoint	Average weighted NPMI over controlled problem-side and idea-side concept pairs. Zero co-occurrence is mapped to -1.	Lower raw values are better; the relative-gain calculation reverses the sign.
Atypicality Z-score	Semantic Scholar paper-count endpoint plus a hypergeometric null	Average weighted $Z = (Observed - Expected)/\sigma$ over the same concept-pair grid.	Lower raw values are better; negative values mean under-combination relative to the null.

Table 11: Implementation details for the four Lens metrics.

Framework	Tokens (K)	Cost ($)
AI-Scientist-v2	82.3	0.231
Idea2Paper	132.7	0.365
AutoResearchClaw	113.9	0.333
ARIS	105.8	0.314
claude-scholar	116.2	0.342

Table 12: Average token use and API cost per CrossSci task for the non-ours baselines across the absolute-judge and pairwise-judge protocols.

DeepSeek-V3.2, while the other entries are direct model runs on the same cases. We report solved-case counts; the accuracy follows directly from the 20-case denominator.

A.11 Social impact

IDEAL is intended to make early-stage scientific ideation more inspectable. By measuring semantic redundancy and literature-level atypicality before experiments are run, the framework can help researchers organize large pools of machine-generated proposals and identify directions that deserve closer expert review.

Support for human judgment. The metrics should be used as screening and analysis tools, not as substitutes for domain expertise. A high Lens score does not establish correctness, feasibility, or scientific value. It indicates that an idea set is less repetitive and less conventional under the measured signals, which can help reviewers decide where to spend attention.

Risks of over-automation. Automatic idea scoring could be misused if diversity or atypicality were treated as sufficient evidence of quality. This risk is especially important in domains where proposed work may affect public safety, environmental policy, health, or infrastructure. In such settings, generated ideas still require literature review, technical validation, and expert oversight before execution.

Access and reproducibility. Scalable evaluation tools may lower the cost of exploring scientific directions, but they also depend on model access, embedding models, NLI models, and literature-search infrastructure. We therefore report the generation settings, structured fields, and metric implementations so that future work can audit, reproduce, or replace individual components.