

# PRISM: Entropy-Guided Multi-View Exploration for Scientific Idea Generation

Jiajun Yu<sup>1,2,\*</sup>, Weifan Mao<sup>2,3,\*</sup>, Kaipeng Zeng<sup>2,4</sup>, Kejia Bian<sup>2,5</sup>  
Zhuo Wang<sup>2,6</sup>, Kun Li<sup>7</sup>, Haishuai Wang<sup>1,†</sup>

<sup>1</sup>Zhejiang Key Lab of Accessible Perception and Intelligent Systems, Zhejiang University

<sup>2</sup>Shanghai Innovation Institute

<sup>3</sup>Shanghai Research Institute for Intelligent Autonomous Systems, Tongji University

<sup>4</sup>School of Computer Science & School of Artificial Intelligence, Shanghai Jiao Tong University

<sup>5</sup>School of Information Science and Electronic Engineering, Shanghai Jiao Tong University

<sup>6</sup>College of Computer Science and Artificial Intelligence, Fudan University

<sup>7</sup>Shanghai AI Lab

## Abstract

Scientific idea generation requires more than fluent continuation from a problem statement: strong ideas often emerge from inspecting the same problem through multiple informative perspectives. We propose PRISM, an entropy-guided multi-view framework for scientific ideation. Given a research problem, PRISM constructs candidate inspection views, estimates their pairwise orthogonality through hypothesis-transition entropy, selects a maximally orthogonal subset, and conditions a backbone model on the selected views to synthesize a single proposal-style idea. We evaluate PRISM on 500 research problems across 23 scientific domains using three generation backbones: Llama-3.1-8B-Instruct, Qwen3-8B, and GPT-5.4. To assess generated ideas, we use two complementary automated judges—a Galactica-based impact discriminator trained from citation-derived scientific preferences and GPT-5.5 blind five-way novelty ranking—as well as a blind human study with five PhD-level evaluators. Across 7,500 generated ideas, PRISM achieves the best impact score and an 80.2% impact pairwise win rate, a 90.9% novelty pairwise win rate, and the highest mean score (3.91/5) in human evaluation. These results suggest that entropy-guided multi-view exploration provides an effective inference-time mechanism for producing more novel and impactful scientific ideas. Our code is available at <https://github.com/jiajunyu1999/PRISM-idea-generation>.

## 1 Introduction

Scientific discovery fundamentally originates with the formulation of novel ideas. Recent large lan-

guage models (LLMs) have demonstrated remarkable potential in reasoning, coding, experimental planning, automated workflows, and biological field (Baek et al., 2025; Lu et al., 2024; Yamada et al., 2025; Liu et al., 2026b; Romera-Paredes et al., 2024; Zhang et al., 2025; Yu et al., 2026; Liang et al., 2025; Li et al., 2026). They can even synthesize research ideas evaluated as highly novel under controlled blind-review settings (Si et al., 2025). However, LLM-generated ideas frequently suffer from inadequate self-evaluation, inconsistent feasibility, and restricted diversity across repeated generations (Si et al., 2025). This tension poses a pivotal question: how can LLM-based ideation systems explore broader conceptual spaces instead of merely generating fluent yet conceptually homogeneous proposals?

A compelling theoretical lens for this challenge is divergent thinking, which emphasizes generating multiple, flexible, and original responses rather than prematurely converging upon a single solution (Guilford, 1950; Torrance, 1974). Recent LLM creativity benchmarks adopt this paradigm, evaluating idea generation across dimensions such as originality, diversity, usefulness, feasibility, fluency, flexibility, and clarity (Ruan et al., 2026; Bellemare-Pepin et al., 2026; Doshi and Hauser, 2024). Collectively, these studies suggest that scientific ideation should be assessed not merely by the terminal quality of a single output, but also by the extent to which the generative process explores a broader range of conceptual directions.

Existing agentic systems increasingly automate end-to-end scientific pipelines, encompassing idea generation, literature grounding, experimental execution, manuscript drafting, and simulated peer

\*Equal contribution.

†Corresponding author: [haishuai.wang@zju.edu.cn](mailto:haishuai.wang@zju.edu.cn)

review (Baek et al., 2025; Lu et al., 2024; Yamada et al., 2025). Parallel initiatives further investigate scientific graph reasoning, mathematical program search, autonomous chemical experimentation, and data-driven discovery benchmarks (Ghafarollahi and Buehler, 2025; Romera-Paredes et al., 2024; Boiko et al., 2023; Chen et al., 2025). Recent multi-agent architectures also demonstrate promise for collaborative scientific ideation (Su et al., 2025). Nevertheless, the majority of existing frameworks rely on repeated prompting, retrieval, or iterative refinement. These mechanisms support local improvement but do not explicitly encourage the generator to inspect the same problem from structurally distinct angles before committing to a proposal.

Concurrently, inference-time methodologies—such as self-consistency, Tree of Thoughts, Self-Refine, and Reflexion—have established that sampling, search, critique, and reflection can substantially enhance LLM reasoning (Wang et al., 2022; Yao et al., 2023; Madaan et al., 2023; Shinn et al., 2023). Multi-agent frameworks further demonstrate that orchestrated cooperation, debate, and role specialization can elevate both task reasoning and evaluation capabilities (Du et al., 2024; Li et al., 2023; Wu et al., 2024; Chan et al., 2024). However, in the context of scientific ideation, diversity must go beyond surface-level variation in wording. A mechanism is needed that steers the model toward genuinely orthogonal perspectives before synthesizing a proposal, so that the final idea reflects a broader search rather than the most obvious continuation of the problem statement.

In this paper, we propose PRISM, an entropy-guided multi-view framework for scientific idea generation. The core observation is that inspecting a research problem from multiple complementary angles before generating a proposal leads to substantially more novel and impactful ideas than direct single-prompt generation. Given a research problem, PRISM constructs a compact set of candidate inspection views (Appendix E), each representing a distinct mode of conceptual exploration. Rather than ranking views independently, PRISM instantiates each candidate view as a task-specific hypothesis and uses a frozen evaluator to compute directional hypothesis-transition entropy between every ordered pair of hypotheses. We symmetrize the two directions to obtain a task-conditioned pairwise orthogonality score. PRISM then selects the size- $k$  subset that maximizes average pairwise orthogonality, favoring complementary views that

are mutually less predictable, and conditions the final generator on the selected views to synthesize a single coherent proposal.

A key design question is whether hypothesis-transition entropy provides a meaningful measure of view orthogonality. We validate this by showing that the transition-entropy-based orthogonality of selected view subsets correlates positively with GPT-judged view orthogonality and that more orthogonal selections lead to stronger novelty and impact scores. This chain of evidence supports pairwise hypothesis-transition entropy as an operational, task-conditioned measure of conceptual distinctiveness for multi-view selection. For evaluation, we construct a 500-problem benchmark from LiveIdeaBench-v2 (Ruan et al., 2024), spanning 23 scientific domains, and compare PRISM against four representative baselines using a Galactica-based impact discriminator (Tong et al., 2026) and GPT-5.5 blind novelty ranking. We further validate results with a blind human study involving five PhD-level researchers from medicine, biology, and artificial intelligence, who score ideas on a 1–5 scale for combined novelty and impact. Our contributions are: (1) the observation that multi-view problem inspection substantially improves idea quality, motivating PRISM as a task-adaptive inference-time framework; (2) a pairwise hypothesis-transition entropy measure and subset-selection objective for task-conditioned view orthogonality; and (3) consistent empirical evidence across 7,500 generated ideas and human expert evaluation that orthogonality-guided multi-view selection improves both novelty and impact over strong baselines.

## 2 Related Work

**Scientific Ideation and Agentic Discovery.** LLM-driven scientific ideation has emerged as a pivotal research frontier, bridging open-ended creativity, hypothesis generation, and automated discovery. While LLMs can generate novel research ideas, they frequently suffer from restricted diversity, inconsistent feasibility, and inadequate self-evaluation (Si et al., 2025; Doshi and Hauser, 2024; Bellemare-Pepin et al., 2026). Recent benchmarks assess these capabilities across diverse dimensions—spanning divergent ideation, context-grounded generation, hypothesis construction, and equation discovery (Ruan et al., 2026; Guo et al., 2025; Qiu et al., 2025; Liu et al., 2025b,a; Ma-

jumder et al., 2025; Shojaee et al., 2025; Yang et al., 2026). Concurrently, agentic frameworks attempt to automate end-to-end scientific workflows, encompassing literature grounding, experimentation, and manuscript drafting (Baek et al., 2025; Lu et al., 2024; Yamada et al., 2025; Gottweis et al., 2025; Schmidgall et al., 2025). Although promising, these systems focus mainly on workflow automation and evaluation, with limited support for systematic exploration of diverse latent ideas.

**Multi-agent Reasoning and Collaborative Generation.** Multi-agent LLM systems are widely utilized for complex reasoning, debate, and collaborative problem-solving (Du et al., 2024; Chen et al., 2026; Wu et al., 2024; Chan et al., 2024). Recent literature further investigates their scaling behaviors, collaboration protocols, and creative potential (Qian et al., 2025; Zhu et al., 2025; Tran et al., 2025; Lin et al., 2025). In the specific context of scientific ideation, frameworks like VirSci and MACC demonstrate that multi-agent collaboration and competitive dynamics can enhance research workflows (Su et al., 2025; Oyama et al., 2026). Diverging from generic role specialization or debate protocols, PRISM selects a mutually orthogonal subset of views using task-conditioned pairwise hypothesis-transition entropy and conditions a single backbone generator on the selected views, systematically steering generation toward complementary conceptual directions rather than relying on repeated sampling or agent-level recombination.

**Evaluation with LLM Judges and Reward Models.** Evaluating open-ended scientific ideation requires scalable paradigms, as traditional exact-match metrics fail to capture novelty, feasibility, or scientific promise. Consequently, LLM-as-a-judge paradigms have become standard for assessing long-form, subjective outputs (Zheng et al., 2023; Liu et al., 2023; Kim et al., 2024). However, recent studies emphasize the necessity of benchmarking the evaluators themselves, given their potential biases, weak reasoning, or misalignment with human preferences (Tan et al., 2025; Bavaresco et al., 2025; Gu et al., 2024; Liu et al., 2026a). Preference-based frameworks and reward model benchmarks offer robust alternatives for scalable candidate comparison (Lambert et al., 2025; Dubois et al., 2023). Notably, in the scientific domain, SciJudgeBench formulates scientific taste via preference modeling to support domain-aware judge training (Tong

et al., 2026). Following this trajectory, our evaluation employs a Galactica-based pointwise reward model trained on SciJudgeBench preferences and GPT-5.5 blind novelty ranking to assess whether entropy-guided multi-view ideation outperforms direct prompting baselines.

### 3 Methodology

#### 3.1 Problem Formulation

We study scientific idea generation from a research-problem statement. Each input instance is represented as a task context  $x$ , which contains a task identifier, a scientific domain, and a concise problem description. A generation method  $G$  maps  $x$  to a single research idea  $y = G(x)$ . The desired output is not a free-form brainstorm list, but a proposal-style idea that states a concrete method, explains its scientific motivation, and describes why it may lead to a novel and impactful research direction.

The key difficulty is that direct prompting often makes the generator follow the most obvious continuation of the problem statement. For example, a problem about expensive simulation may immediately trigger generic surrogate modeling, while a problem about multimodal measurement may trigger generic fusion. We therefore formulate ideation as multi-view exploration: before generating the final idea, the model should inspect the same problem from multiple perspectives and then select a mutually complementary subset whose induced hypotheses are maximally orthogonal for the current instance.

#### 3.2 PRISM: Entropy-Guided Multi-View Exploration

PRISM consists of three modules: multi-view inspection, transition-entropy-based orthogonal view selection, and view-conditioned idea generation. The first module defines the conceptual space from which the problem can be inspected. The second module selects a mutually orthogonal subset of views for the given problem using pairwise hypothesis-transition entropy. The third module synthesizes these views into a single coherent idea.

**Candidate view construction.** Given a task context  $x$ , PRISM constructs a candidate view pool

$$\mathcal{V} = v_{i=1}^M, \quad (1)$$

where each  $v_i$  represents a distinct reasoning perspective, such as a mechanism, abstraction level,

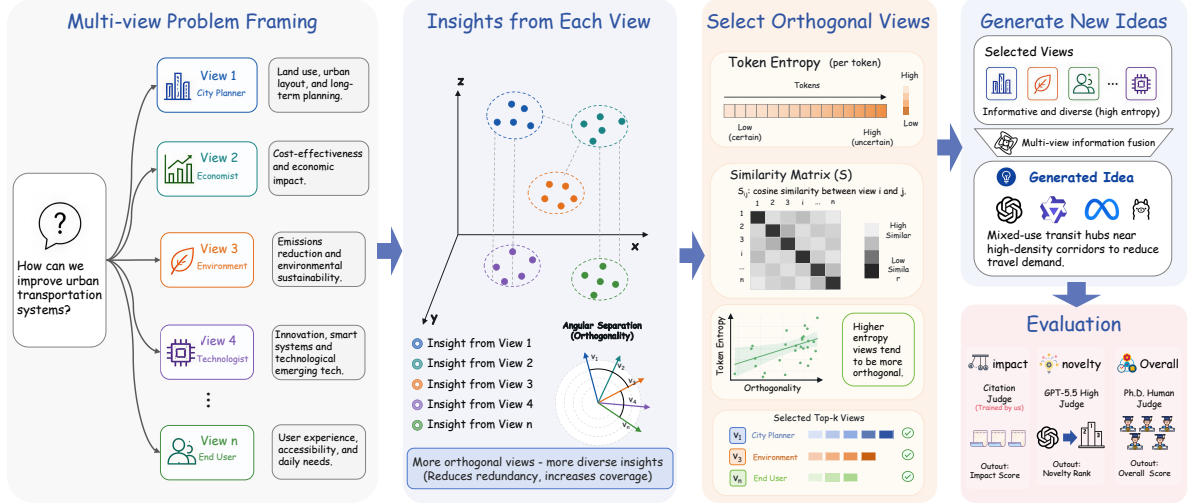


Figure 1: Overview of PRISM. Given a research problem, PRISM constructs a multi-view inspection space, instantiates candidate views as task-specific hypotheses, estimates pairwise view orthogonality through symmetric hypothesis-transition entropy, selects the size- $k$  subset maximizing average pairwise orthogonality, and conditions a backbone model on the selected views to synthesize a single proposal-style idea.

evidence source, or modeling assumption. These views are treated as candidate directions without independent informativeness scores.

**View-conditioned hypothesis generation.** To assess semantic relations between abstract views, we instantiate each view as a task-specific hypothesis:

$$h_i = r_\psi(x, v_i), \quad i = 1, \dots, M, \quad (2)$$

forming

$$\mathcal{H} * x = h_i * i = 1^M. \quad (3)$$

These hypotheses act as semantic probes: two views are redundant if their hypotheses are easily transformed into one another, and orthogonal if such transformation is uncertain.

**Pairwise hypothesis-transition entropy.** We estimate the relationship between hypotheses using a frozen open-source language model  $q_\theta$ , where  $\theta$  denotes its fixed model parameters. For every ordered pair of distinct hypotheses  $(h_i, h_j)$ , the model is given the task context  $x$ , the source hypothesis  $h_i$ , and a fixed transition instruction  $\pi_{tr}$  that asks it to reason toward an alternative hypothesis for the same problem. The target hypothesis  $h_j$  is then evaluated autoregressively under this transition context.

Let

$$h_j = (z_{j,1}, \dots, z_{j,L_j}) \quad (4)$$

denote the token sequence of the target hypothesis, where  $L_j$  is its number of tokens,  $z_{j,t}$  is its  $t$ -th

token, and  $z_{j,<t} = (z_{j,1}, \dots, z_{j,t-1})$  is the token prefix preceding position  $t$ . At each position, the evaluator produces the next-token distribution

$$p_{i \rightarrow j,t}(\cdot) = q_\theta(\cdot \mid \pi_{tr}, x, h_i, z_{j,<t}). \quad (5)$$

The directional transition entropy from  $h_i$  to  $h_j$  is defined as

$$E_{i \rightarrow j} = \frac{1}{L_j} \sum_{t=1}^{L_j} H(p_{i \rightarrow j,t}), \quad (6)$$

where  $H(\cdot)$  is the Shannon entropy of a token distribution. More explicitly,

$$H(p) = - \sum_{w \in \Omega} p(w) \log p(w), \quad (7)$$

where  $\Omega$  denotes the vocabulary of the evaluator model. A low value of  $E_{i \rightarrow j}$  indicates that the evaluator can move from the reasoning represented by  $h_i$  toward  $h_j$  with relatively confident token predictions. This suggests that the two hypotheses, and therefore their underlying views, share substantial structure. In contrast, a high transition entropy indicates that the target hypothesis is difficult to reach predictably from the source hypothesis, suggesting a more distinct reasoning direction.

Because the transition entropy is directional,  $E_{i \rightarrow j}$  need not equal  $E_{j \rightarrow i}$ . We therefore define a symmetric pairwise orthogonality score

$$D_{ij} = \frac{1}{2} (E_{i \rightarrow j} + E_{j \rightarrow i}), \quad D_{ii} = 0. \quad (8)$$

Here,  $D_{ij}$  measures the task-conditioned orthogonality between views  $v_i$  and  $v_j$  through their induced hypotheses. A larger  $D_{ij}$  means that the two views induce hypotheses that are less mutually predictable. We use  $D_{ij}$  as an operational measure of view orthogonality; it is not required to satisfy all properties of a mathematical distance metric.

**Top- $k$  orthogonal view selection.** Rather than ranking views independently, PRISM selects a subset whose members are mutually orthogonal. Let  $k$  denote the number of views to retain, with  $2 \leq k \leq M$ . The selected view set is defined as

$$\mathcal{V}_x^* = \operatorname{argmax}_{\substack{\mathcal{S} \subseteq \mathcal{V} \\ |\mathcal{S}|=k}} \frac{2}{k(k-1)} \sum_{\substack{v_i, v_j \in \mathcal{S} \\ i < j}} D_{ij}, \quad (9)$$

where  $\mathcal{S}$  denotes a candidate subset of views and  $\mathcal{V}_x^*$  denotes the selected subset for task  $x$ . The objective maximizes the average pairwise transition entropy among the selected views. Consequently, it favors a collection of complementary reasoning directions rather than several views that are individually unusual but mutually redundant. When exhaustive subset search is impractical, the objective can be approximated using greedy farthest-first selection.

**Multi-view idea generation.** Finally, the selected views are provided to a backbone generator  $g_\phi$  together with the original task context:

$$y = g_\phi(x, \mathcal{V}_x^*), \quad (10)$$

where  $g_\phi$  denotes the final idea-generation model and  $y$  is the resulting proposal. The intermediate hypotheses  $\mathcal{H}_x$  are used to estimate relationships among the candidate views; the final generator is conditioned on the selected views themselves. The generation prompt requires the model to synthesize these views into a single coherent method grounded in the original problem, rather than presenting them as disconnected sections. The hypothesis generator  $r_\psi$  and final generator  $g_\phi$  may be implemented using the same backbone model with different prompts.

### 3.3 Impact Discriminator Modeling

We formulate impact judging as pointwise reward modeling. Given a generated idea  $x$ , the judge predicts a scalar score

$$s_\theta(x) \in \mathbb{R} \quad (11)$$

where larger values indicate higher expected idea impact. Although training uses pairwise supervision from SciJudgeBench (Tong et al., 2026), the learned model can directly score, rank, and rerank individual candidate ideas.

**Preference learning.** Each training instance contains a preferred idea  $x^+$  and a non-preferred idea  $x^-$ . We optimize a Bradley–Terry pairwise objective (Bradley and Terry, 1952):

$$\mathcal{L}_{\text{pair}} = -\log \sigma(s_\theta(x^+) - s_\theta(x^-)) \quad (12)$$

where  $\sigma(\cdot)$  denotes the logistic sigmoid function.

**Input and scoring interface.** Each idea is represented using a unified structured template containing its field, subfield, title, and abstract. Citation counts and ground-truth labels are excluded from the input to avoid information leakage. For both encoder-style and decoder-only backbones, the scoring function is implemented as

$$s_\theta(x) = \text{RewardHead}(\text{Pool}(\text{Backbone}(x))) \quad (13)$$

where Pool denotes the pooled sequence representation for encoder models or the last non-padding token representation for decoder-only models. The same scalar scores induce pairwise preferences by

$$p(x_a \succ x_b) = \sigma(s_\theta(x_a) - s_\theta(x_b)). \quad (14)$$

Thus, a single judge can score an isolated idea or compare two candidates without changing the underlying model.

**Weighted pairwise training.** To emphasize clearer preference pairs while preventing ambiguous pairs from dominating optimization, we optionally define

$$\tilde{w} = \alpha \log(1 + \Delta_{\text{abs}}) + \beta \Delta_{\text{rel}} \quad (15)$$

and clip it as

$$w = \min(w_{\text{max}}, \max(w_{\text{min}}, \tilde{w})) \quad (16)$$

Here  $\Delta_{\text{abs}}$  and  $\Delta_{\text{rel}}$  denote the absolute and relative citation gaps,  $\alpha$  and  $\beta$  are scaling coefficients, and  $w_{\text{min}}, w_{\text{max}}$  bound the final pair weight. The weighted objective is

$$\mathcal{L}_{\text{weighted}} = w \cdot [-\log \sigma(s_\theta(x^+) - s_\theta(x^-))] \quad (17)$$

and we further apply

$$\mathcal{L}_{\text{reg}} = \lambda (\mathbb{E}[s_\theta(x^+)^2] + \mathbb{E}[s_\theta(x^-)^2]) \quad (18)$$

The coefficient  $\lambda$  controls score-magnitude regularization, while detailed weighting choices are deferred to Appendix B.

**Training and evaluation.** We train the reward model with optional citation-gap-based loss weighting and lightweight score regularization, while using citation-derived statistics only for training weights and offline analysis. Pairwise accuracy under the training-time validation protocol is used for checkpoint selection, and final evaluation is conducted on the in-domain test split of SciJudgeBench. Detailed input construction, weighting objectives, implementation choices, and evaluation metrics are provided in Appendix B.

## 4 Experiments

### 4.1 Experimental Setup

We organize the evaluation around two connected questions: whether PRISM improves generated ideas across different generation backbones, and whether the impact discriminator used for scoring provides a reliable scientific-preference signal.

**Benchmark.** To evaluate scientific idea generation, we construct a 500-problem benchmark from LiveIdeaBench-v2 (Ruan et al., 2024). The benchmark spans 23 scientific domains and preserves heterogeneous problem formulations across disciplines. Each sample in LiveIdeaBench-v2 contains a free-form idea field whose surface form varies substantially. We therefore use an LLM to extract the research problem from each sample, deduplicate normalized problem statements, remove low-quality extractions with heuristic filters and sample 500 roughly domain-balanced problems. The construction details can be found in Appendix A.

**Generation methods.** We instantiate every idea-generation method on three backbones: Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Qwen3-8B (Yang et al., 2025), and GPT-5.4. This setting tests whether a method improves idea generation for both open-weight instruction models and a stronger closed-source backbone. We compare PRISM against four representative scientific-ideation baselines: AI Scientist (Lu et al., 2024), AI Researcher (Tang et al., 2026), ResearchAgent (Baek et al., 2025), and VirSci (Su et al., 2025). All methods produce one final idea per problem per backbone. For fair comparison, we disable RAG retrieval for all methods, so no method uses external databases during idea generation. All outputs are normalized as single abstract-style idea paragraphs and evaluated under the same judge protocols.

**Evaluation setting.** We evaluate ideas from three perspectives: impact, novelty, and human judgment. Impact is measured by our Galactica-based impact discriminator, and we report the mean score and pairwise win rate. Novelty is evaluated by GPT-5.5 blind ranking, where the judge ranks five anonymized ideas from the five methods for each backbone-problem pair. We report mean rank, top-1 rate, and pairwise win rate. For human evaluation, one human evaluator independently reviews ideas generated on 100 problems using the GPT-5.4 backbone. For each problem, the evaluator is presented with five anonymized ideas, one from each method, and assigns each idea a score from 1 to 5.

**Impact judge data.** Because the impact metric is produced by a learned discriminator, we separately validate this judge before using it in the main evaluation. We use SciJudgeBench (Tong et al., 2026) as the primary benchmark for training and validation. Models are trained on pairwise preference data constructed from the official training split, while checkpoint selection follows the validation protocol implemented in the training pipeline. Evaluation is conducted on the test split. The full training details are provided in Appendix B.

### 4.2 Main Results

Table 1 presents the main results. PRISM obtains the highest impact score, top-1 rate, and win rate across all backbones, with an overall impact top-1 rate of 57.3% and a pairwise win rate of 80.2%. The difference is larger on the two open-weight backbones, where PRISM reaches 85.3% on Llama-3.1-8B and 84.3% on Qwen3-8B. These results suggest that multi-view guidance is not limited to a strong closed-source generator and can also help smaller instruction models in this evaluation. The novelty ranking shows a similar pattern. PRISM obtains an overall mean rank of 1.36 and is ranked first in 78.4% of the 1,500 backbone-problem comparisons. Its pairwise novelty win rate reaches 90.9%, while the strongest baseline remains below 44.0%. The trend is consistent across all three backbones, though the margin is smaller on Qwen3-8B.

### 4.3 Human Expert Evaluation

We conduct a blind human study with five PhD-level researchers from medicine (1), biology (1), and artificial intelligence (3), with each question evaluated by one researcher in a relevant field. The researchers were recruited through personal aca-

| Impact Evaluation: Galactica-Based Scientific Reward Model |                  |                  |               |                  |                  |               |                  |                  |               |                  |                  |               |
|------------------------------------------------------------|------------------|------------------|---------------|------------------|------------------|---------------|------------------|------------------|---------------|------------------|------------------|---------------|
| Method                                                     | Overall          |                  |               | GPT-5.4          |                  |               | Llama-3.1-8B     |                  |               | Qwen3-8B         |                  |               |
|                                                            | Score $\uparrow$ | Top-1 $\uparrow$ | WR $\uparrow$ | Score $\uparrow$ | Top-1 $\uparrow$ | WR $\uparrow$ | Score $\uparrow$ | Top-1 $\uparrow$ | WR $\uparrow$ | Score $\uparrow$ | Top-1 $\uparrow$ | WR $\uparrow$ |
| PRISM                                                      | <b>-0.457</b>    | <b>57.3</b>      | <b>80.2</b>   | <b>0.286</b>     | <b>41.6</b>      | <b>70.9</b>   | <b>-1.142</b>    | <b>67.6</b>      | <b>85.3</b>   | <b>-0.516</b>    | <b>62.6</b>      | <b>84.3</b>   |
| ResearchAgent                                              | -1.178           | 12.9             | 46.3          | 0.007            | 21.8             | 55.2          | -2.159           | 6.6              | 40.4          | -1.382           | 10.2             | 43.1          |
| AI Researcher                                              | -1.197           | 10.9             | 44.8          | -0.139           | 12.0             | 44.9          | -2.060           | 10.4             | 46.4          | -1.391           | 10.2             | 43.0          |
| AI Scientist                                               | -1.251           | 10.1             | 40.6          | -0.256           | 14.0             | 39.1          | -2.119           | 7.8              | 40.2          | -1.377           | 8.6              | 42.5          |
| VirSci                                                     | -1.348           | 8.9              | 38.1          | -0.253           | 10.6             | 39.9          | -2.223           | 7.6              | 37.6          | -1.567           | 8.4              | 36.9          |

| Novelty Evaluation: GPT-5.5 Blind Five-Way Ranking |                   |                  |               |                   |                  |               |                   |                  |               |                   |                  |               |
|----------------------------------------------------|-------------------|------------------|---------------|-------------------|------------------|---------------|-------------------|------------------|---------------|-------------------|------------------|---------------|
| Method                                             | Overall           |                  |               | GPT-5.4           |                  |               | Llama-3.1-8B      |                  |               | Qwen3-8B          |                  |               |
|                                                    | Rank $\downarrow$ | Top-1 $\uparrow$ | WR $\uparrow$ | Rank $\downarrow$ | Top-1 $\uparrow$ | WR $\uparrow$ | Rank $\downarrow$ | Top-1 $\uparrow$ | WR $\uparrow$ | Rank $\downarrow$ | Top-1 $\uparrow$ | WR $\uparrow$ |
| PRISM                                              | <b>1.36</b>       | <b>78.4</b>      | <b>90.9</b>   | <b>1.14</b>       | <b>89.6</b>      | <b>96.4</b>   | <b>1.18</b>       | <b>88.4</b>      | <b>95.5</b>   | <b>1.77</b>       | <b>57.2</b>      | <b>80.8</b>   |
| ResearchAgent                                      | 3.76              | 2.5              | 31.0          | 3.83              | 0.6              | 29.2          | 3.91              | 1.2              | 27.4          | 3.54              | 5.6              | 36.5          |
| AI Researcher                                      | 3.33              | 6.7              | 41.8          | 3.15              | 4.2              | 46.2          | 3.61              | 2.4              | 34.8          | 3.22              | 13.4             | 44.4          |
| AI Scientist                                       | 3.31              | 6.3              | 42.4          | 3.69              | 2.0              | 32.8          | 2.94              | 5.6              | 51.4          | 3.28              | 11.4             | 43.0          |
| VirSci                                             | 3.24              | 6.1              | 44.0          | 3.18              | 3.6              | 45.5          | 3.36              | 2.4              | 41.0          | 3.18              | 12.4             | 45.5          |

Table 1: Main evaluation results for impact and novelty. For impact, Score is the mean scalar score from the Galactica-based scientific reward model; for novelty, Rank is the mean position assigned by GPT-5.5 blind ranking. Top-1 is the percentage of backbone-problem groups where the method obtains the best score or rank; WR is the pairwise win rate (%) against the other four methods within the same backbone-problem group.

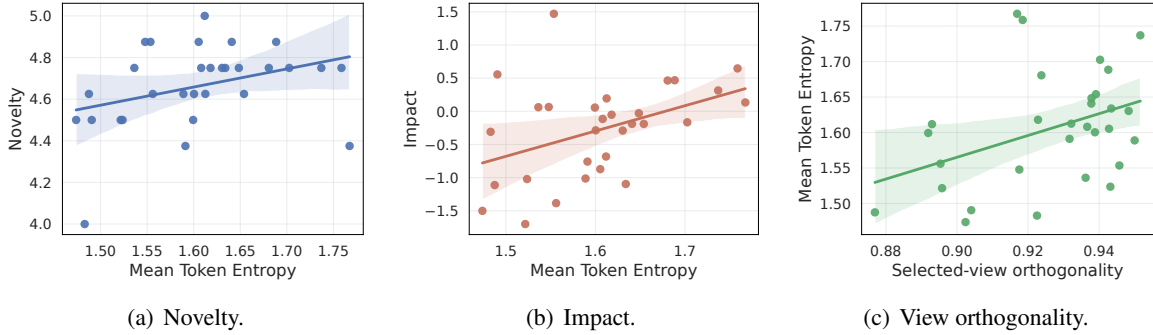


Figure 2: Panels (a) and (b) plot mean token entropy of the selected views against novelty and impact scores, respectively; both show a positive trend, indicating that higher-entropy view selections lead to more novel and impactful ideas. Panel (c) plots mean token entropy against GPT-judged view orthogonality, confirming that token entropy is a reliable proxy for view distinctiveness. Lines show linear trends with 95% confidence intervals.

demic contacts and were not paid for their participation. Evaluators participated voluntarily and were informed that their anonymized scores would be used for research evaluation. Each evaluator scores ideas on a 1–5 integer scale reflecting combined *novelty* and *impact*, without knowledge of which method produced each idea. Table 2 reports human scores alongside the automated impact and novelty rankings for the same GPT-5.4 backbone, enabling direct cross-signal comparison. PRISM ranks first on all three evaluation axes. The per-question Spearman correlation between human scores and GPT-5.5 novelty rankings is  $\rho = 0.36$  ( $p < 0.001$ ), while the correlation with Galactica impact scores is weaker at  $\rho = 0.05$  ( $p < 0.05$ ), sug-

gesting that human evaluators align more closely with the originality signal than with the citation-derived impact proxy. A representative case is shown in Appendix D.

#### 4.4 Analysis

**Analysis of the impact judge model.** Table 4 reports results on the SciJudgeBench test split. JudgeReward models consistently outperform both SciJudge reference judges. JudgeReward-Galactica-6.7B achieves the best overall accuracy (89.5%), exceeding SciJudge-Qwen2.5-32B and SciJudge-Qwen3-30B by 5.8 and 8.9 percentage points, respectively, and performs best on CS, Physics, and Others. JudgeReward-Qwen2.5-7B-

| Method        | Human            |          | Impact           |          | Novelty           |          |
|---------------|------------------|----------|------------------|----------|-------------------|----------|
|               | Score $\uparrow$ | Rank     | Score $\uparrow$ | Rank     | Rank $\downarrow$ | Rank     |
| PRISM         | <b>3.91</b>      | <b>1</b> | <b>-0.457</b>    | <b>1</b> | <b>1.36</b>       | <b>1</b> |
| AI Researcher | 3.50             | 2        | -1.197           | 3        | 3.33              | 4        |
| VirSci        | 3.40             | 3        | -1.348           | 5        | 3.24              | 2        |
| AI Scientist  | 3.34             | 4        | -1.251           | 4        | 3.31              | 3        |
| ResearchAgent | 2.93             | 5        | -1.178           | 2        | 3.76              | 5        |

Per-question Spearman: Human–Novelty  $\rho=0.36^{***}$ ; Human–Impact  $\rho=0.05^*$

Table 2: Human expert evaluation alongside automated rankings (GPT-5.4 backbone). Human Score: mean 1–5 rating from five PhD-level evaluators (blind). Impact Score: mean Galactica reward score. Novelty Rank: mean GPT-5.5 blind rank (1=best). Rank columns show ordinal position across the five methods. Per-question Spearman correlations are computed over all method–problem pairs;  $^*p<0.05$ ,  $^{***}p<0.001$ .

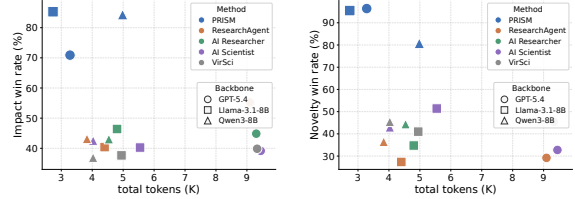
| Family                       | Model          | Acc. $\uparrow$ |
|------------------------------|----------------|-----------------|
| SciJudge (Tong et al., 2026) | Qwen2.5-32B    | 83.7            |
|                              | Qwen3-30B      | 80.6            |
| JudgeReward                  | SciBERT-0.1B   | 81.3            |
|                              | SciTulu-7B     | 84.0            |
|                              | Qwen2.5-7B     | 88.7            |
|                              | Llama-3.1-8B   | 89.4            |
| Our Impact Judge Model       | Galactica-6.7B | <b>89.5</b>     |

Table 3: Auxiliary validation of scientific impact judges on the SciJudgeBench test split. Acc. is pairwise accuracy (%); the shaded row denotes the Galactica-based discriminator used for the main impact evaluation.

Instruct and JudgeReward-Llama-3.1-8B-Instruct also achieve strong accuracies of 88.7% and 89.4%. In contrast, JudgeReward-SciBERT-0.1B underperforms, highlighting the importance of model capacity. Overall, these results demonstrate the effectiveness of pairwise scientific preference supervision for training pointwise reward models to evaluate scientific idea impact.

**Analysis of main results.** Table 1 shows that impact and novelty capture different signals: their winners agree in only 47.2% of cases, with weak score-rank correlation ( $\rho = 0.07$ ,  $p<0.001$ ). Impact gains are strongest for open-weight backbones, reaching 85.3% on Llama-3.1-8B and 84.3% on Qwen3-8B, suggesting that selected views provide useful planning scaffolds. Novelty wins are also not merely impact wins: PRISM wins impact in 859 groups and novelty in 1,176 groups, but wins both in only 668 groups.

**Analysis of selected-view orthogonality.** Figures 2(a) and 2(b) plot mean token entropy of the selected views against novelty and impact, respectively. Both trends are positive: problems where



(a) Impact win rate.

(b) Novelty win rate.

Figure 3: Token cost and win rate. Each point is one method-backbone pair. Color denotes method, marker shape denotes backbone, the x-axis is average total token usage, and the y-axis is pairwise win rate under the corresponding judge.

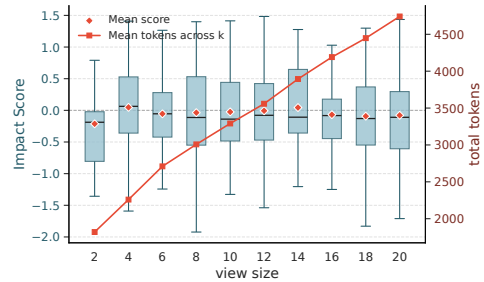


Figure 4: View-count sweep used to choose the main experimental setting. For each initial view pool size  $N$ , the plot reports Galactica raw scores for the best selection size  $k$  and the corresponding estimated total token cost. Four views provide a strong score with substantially lower cost than larger pools.

the selected views have higher mean token entropy tend to receive higher novelty rankings and higher impact scores. This suggests that entropy-guided view selection steers the generator toward conceptually less obvious directions that are recognized as more novel and impactful by both judges. Figure 2(c) shows a positive correlation between transition-entropy-based orthogonality and GPT-judged view orthogonality, supporting hypothesis-transition entropy as a useful measure of view distinctiveness. Table 5 further provides an auxiliary 500-problem pilot (GPT-5.4, 8-view pool, select 3), where high-entropy selection achieves a 53.6% pairwise impact win rate over low-entropy selection. Together, these results suggest that entropy-based signals are predictive of both view distinctiveness and downstream idea quality.

**Analysis of the token cost.** We further compare total token usage with impact and novelty win rates for each method-backbone pair. Figure 3 shows that PRISM remains in the high-win-rate region while using fewer tokens than the GPT-5.4 base-

| Model                                    | CS $\uparrow$ | Math $\uparrow$ | Physics $\uparrow$ | Others $\uparrow$ | Overall $\uparrow$ |
|------------------------------------------|---------------|-----------------|--------------------|-------------------|--------------------|
| SciJudge-Qwen2.5-32B (Tong et al., 2026) | 85.4          | 77.9            | 82.2               | 89.9              | 83.7               |
| SciJudge-Qwen3-30B (Tong et al., 2026)   | 83.5          | 78.7            | 78.7               | 82.3              | 80.6               |
| JudgeReward-Qwen2.5-7B-Instruct          | 91.2          | <b>92.0</b>     | 85.1               | 88.5              | 88.7               |
| JudgeReward-Llama-3.1-8B-Instruct        | 92.3          | 90.9            | 86.1               | 89.8              | 89.4               |
| JudgeReward-SciBERT-0.1B                 | 83.5          | 82.6            | 79.5               | 80.4              | 81.3               |
| JudgeReward-SciTulu-7B                   | 87.1          | 85.2            | 81.5               | 83.1              | 84.0               |
| JudgeReward-Galactica-6.7B               | <b>91.4</b>   | 90.3            | <b>87.4</b>        | <b>90.1</b>       | <b>89.5</b>        |

Table 4: Full Results of scientific impact judges on the SciJudgeBench test split. Acc. is pairwise accuracy (%); the shaded row denotes the Galactica-based discriminator used for the main impact evaluation.

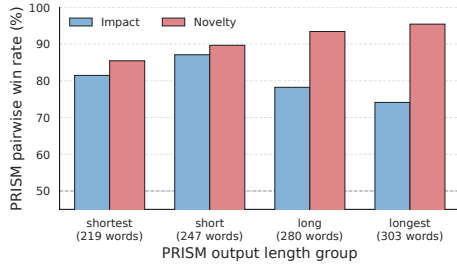


Figure 5: PRISM pairwise win rate grouped by output length. Novelty win rate increases with longer outputs, while impact win rate does not show the same monotonic trend.

| Condition                 | Impact $\uparrow$ | Nov. Rank $\downarrow$ | Impact WR $\uparrow$ |
|---------------------------|-------------------|------------------------|----------------------|
| Low-Entropy Top-3         | -0.522            | 4.12                   | 46.4%                |
| High-Entropy Top-3 (ours) | <b>-0.475</b>     | <b>3.65</b>            | <b>53.6%</b>         |

Table 5: High- vs. low-entropy view selection (GPT-5.4, 8-view pool). Impact WR is the pairwise win rate of each condition against the other.

lines and a moderate token budget for the open-weight backbones. Qwen3-8B is a useful exception to the simplest token-cost interpretation: several baselines use fewer tokens, and the novelty margin is smaller than on the other two backbones, likely because this backbone more often produces generic or less tightly integrated proposals under the same prompt format. Even in this setting, PRISM still obtains the highest impact and novelty win rates, suggesting that the observed gains are not simply explained by spending more generation tokens.

**Analysis of the number of views.** Figure 4 presents a sweep on 10 GPT-5.4 tasks, where each initial pool size  $N$  is evaluated with its best selection size  $k$  under the Galactica impact judge. Performance improves substantially from two to four views, but larger pools lead to non-monotonic gains and higher token cost. Although the highest raw score is obtained with a larger pool, its

advantage over four views is marginal relative to the added cost and selection complexity. We therefore use four views as a robust default rather than a theoretically optimal configuration.

**Analysis of the output length.** We further test whether the gains mainly come from longer ideas. We divide PRISM outputs into four word-count groups and compute pairwise win rates against the four baselines. As shown in Figure 5, novelty win rate increases from 85.4% to 95.4%, indicating a mild length bias, while impact shows no such trend. Since output lengths are comparable across methods, this bias is largely controlled.

## 5 Conclusion

We presented PRISM, an entropy-guided multi-view exploration framework for scientific idea generation. PRISM reframes ideation from direct single-prompt continuation into a task-adaptive inspection process: it constructs multiple views of a research problem, estimates their pairwise orthogonality through symmetric hypothesis-transition entropy, selects a complementary subset by maximizing average pairwise orthogonality, and conditions the generator on the selected views to synthesize a grounded proposal. Across 500 research problems, 23 scientific domains, three generation backbones, and 7,500 generated ideas, PRISM achieves the strongest impact scores under a Galactica discriminator and the best novelty rankings under blind GPT-5.5 comparison, with analyses showing that the gains are not explained by output length, token cost, or a single evaluator signal. These results suggest that entropy-guided multi-view exploration is an inference-time primitive for making LLM-based scientific ideation more adaptive and diverse.

## Limitations

PRISM has several limitations. The estimated view orthogonality depends on both the hypoth-

esis generator and the frozen transition evaluator, and stronger or domain-specific choices for these components may improve robustness. In addition, while our human evaluation spans multiple scientific domains, broader coverage and more extensive agreement analysis would further strengthen the study. Finally, PRISM introduces additional inference overhead from pairwise hypothesis-transition scoring and subset selection.

## 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.
- Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, and 1 others. 2025. Llm instead of human judges? a large scale empirical study across 20 nlp evaluation tasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 238–255.
- Antoine Bellemare-Pepin, François Lespinasse, Philipp Thölke, Yann Harel, Kory Mathewson, Jay A Olson, Yoshua Bengio, and Karim Jerbi. 2026. Divergent creativity in humans and large language models. *Scientific Reports*, 16(1):1279.
- Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. 2023. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578.
- Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.
- Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2024. Chateval: Towards better llm-based evaluators through multi-agent debate. In *International conference on learning representations*, volume 2024, pages 9079–9093.
- Huangwei Chen, Wu Li, Junhao Jia, Yining Chen, Xiaotao Pang, Ya-Long Chen, Li Gonghui, Haishuai Wang, Jiajun Bu, and Lei Wu. 2026. Beyond the individual: Virtualizing multi-disciplinary reasoning for clinical intake via collaborative agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 16074–16101.
- Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, and 1 others. 2025. Scienceagent-bench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations*, volume 2025, pages 96934–96990.
- DeepSeek-AI. 2026. Deepseek-v4: Towards highly efficient million-token context intelligence.
- Anil R Doshi and Oliver P Hauser. 2024. Generative ai enhances individual creativity but reduces the collective diversity of novel content. *Science advances*, 10(28):eadn5290.
- Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first international conference on machine learning*.
- Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2023. AlpacaFarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36:30039–30069.
- 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*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, and 1 others. 2024. A survey on llm-as-a-judge. *The Innovation*.
- Joy Paul Guilford. 1950. Creativity. *American psychologist*, 5(9):444.
- 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.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

- Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353.
- Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, and 1 others. 2025. Rewardbench: Evaluating reward models for language modeling. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1755–1797.
- Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for "mind" exploration of large language model society. *Advances in neural information processing systems*, 36:51991–52008.
- Shiyu Li, Ziqi Yan, Zhihao Wu, Jielong Lu, Weiran Liao, Jiajun Yu, Genjie Li, Zeyu Chu, Jiajun Bu, and Haishuai Wang. 2026. Deepbd: A grounded agentic workflow for variant prioritization and diagnosis of genetic birth defects. *arXiv preprint arXiv:2606.24779*.
- Yaozhen Liang, Xiao Liu, Jiajun Yu, Zhouhua Fang, Qunsheng Zou, Linghan Zheng, Yong Li, Zhiwei Liu, and Haishuai Wang. 2025. Explain: Enhancing retrieval-augmented generation with entity summary. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 1520–1529.
- Yi-Cheng Lin, Kang-Chieh Chen, Zhe-Yan Li, Tzu-Heng Wu, Tzu-Hsuan Wu, Kuan-Yu Chen, Hung-yi Lee, and Yun-Nung Chen. 2025. Creativity in llm-based multi-agent systems: A survey. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27572–27595.
- Haokun Liu, Sicong Huang, Jingyu Hu, Yangqiaoyu Zhou, and Chenhao Tan. 2025a. Hypobench: Towards systematic and principled benchmarking for hypothesis generation. *arXiv preprint arXiv:2504.11524*.
- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 2511–2522.
- Yixin Liu, Yue Yu, DiJia Su, Sid Wang, Xuwei Wang, Song Jiang, Bo Liu, Arman Cohan, Yuandong Tian, and Zhengxing Chen. 2026a. Examining reasoning llms-as-judges in non-verifiable llm post-training. *arXiv preprint arXiv:2603.12246*.
- Yujie Liu, Zonglin Yang, Tong Xie, Jinjie Ni, Ben Gao, Yuqiang Li, Shixiang Tang, Wanli Ouyang, Erik Cambria, and Dongzhan Zhou. 2025b. Research-bench: Benchmarking llms in scientific discovery via inspiration-based task decomposition. *arXiv preprint arXiv:2503.21248*.
- Zihang Liu, Fang Zhouhua, Hui Liu, Zhiwei Liu, Yong Li, and Haishuai Wang. 2026b. Rfs-guard: Detecting reasoning hallucinations via cross-phase routing focus in large reasoning models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19371–19385.
- 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*.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594.
- Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. 2025. Discoverybench: Towards data-driven discovery with large language models. In *International Conference on Learning Representations*, volume 2025, pages 4556–4579.
- Satoshi Oyama, Yuko Sakurai, and Hisashi Kashima. 2026. Macc: Multi-agent collaborative competition for scientific exploration. *arXiv preprint arXiv:2603.03780*.
- Chen Qian, Zihao Xie, Yifei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, and 1 others. 2025. Scaling large language model-based multi-agent collaboration. In *International Conference on Learning Representations*, volume 2025, pages 41488–41505.
- Yansheng Qiu, Haoquan Zhang, Zhaopan Xu, Ming Li, Diping Song, Zheng Wang, and Kaipeng Zhang. 2025. Ai idea bench 2025: Ai research idea generation benchmark. *arXiv preprint arXiv:2504.14191*.
- Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, and 1 others. 2024. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475.
- Kai Ruan, Xuan Wang, Jixiang Hong, and Hao Sun. 2024. Liveideabench: Evaluating llms’ scientific creativity and idea generation with minimal context. *arXiv e-prints*, pages arXiv–2412.
- Kai Ruan, Xuan Wang, Jixiang Hong, Peng Wang, Yang Liu, and Hao Sun. 2026. Evaluating llms’ divergent

- thinking capabilities for scientific idea generation with minimal context. *Nature communications*.
- Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. 2025. Agent laboratory: Using llm agents as research assistants. *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 5977–6043.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652.
- Parshin Shojaei, Ngoc-Hieu Nguyen, Kazem Meidani, Amir Barati Farimani, Khoa D Doan, and Chandan K Reddy. 2025. Llm-srbench: A new benchmark for scientific equation discovery with large language models. *arXiv preprint arXiv:2504.10415*.
- Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. 2025. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers. In *International Conference on Learning Representations*, volume 2025, pages 94003–94092.
- Haoyang Su, Renqi Chen, Shixiang Tang, Zhenfei Yin, Xinzhe Zheng, Jinzhe Li, Biqing Qi, Qi Wu, Hui Li, Wanli Ouyang, and 1 others. 2025. Many heads are better than one: Improved scientific idea generation by a llm-based multi-agent system. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28201–28240.
- Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Tang, Alejandro Cuadron, Chenguang Wang, Raluca Popa, and Ion Stoica. 2025. Judgebench: A benchmark for evaluating llm-based judges. In *International Conference on Learning Representations*, volume 2025, pages 63277–63303.
- Jiabin Tang, Lianghao Xia, Zhonghang Li, and Chao Huang. 2026. Ai-researcher: Autonomous scientific innovation. *Advances in Neural Information Processing Systems*, 38:9481–9520.
- Jingqi Tong, Mingzhe Li, Hangcheng Li, Yongzhuo Yang, Yurong Mou, Weijie Ma, Zhiheng Xi, Hongji Chen, Xiaoran Liu, Qinyuan Cheng, and 1 others. 2026. Ai can learn scientific taste. *arXiv preprint arXiv:2603.14473*.
- E Paul Torrance. 1974. Torrance tests of creative thinking. *Educational and psychological measurement*.
- Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O’Sullivan, and Hoang D Nguyen. 2025. Multi-agent collaboration mechanisms: A survey of llms. *arXiv preprint arXiv:2501.06322*.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.
- Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. 2024. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First conference on language modeling*.
- 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*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Zonglin Yang, Wanhao Liu, Ben Gao, Yujie Liu, Wei Li, Tong Xie, Lidong Bing, Wanli Ouyang, Erik Cambria, and Dongzhan Zhou. 2026. Moose-chem2: Exploring llm limits in fine-grained scientific hypothesis discovery via hierarchical search. *Advances in Neural Information Processing Systems*, 38:89045–89076.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.
- Jiajun Yu, Linghan Zheng, Hui Liu, Jiayuan Dong, Yue Shen, Zhiwei Liu, Yaozhen Liang, Yong Li, and Haishuai Wang. 2026. Code-based english models reveal surprising performance on chinese qa pair extraction task. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 5060–5066.
- Tianyun Zhang, Jiajun Yu, Shang Lou, Yaozhen Liang, Yue Liang, Zhekai Li, Haishuai Wang, Shanshan Pei, and Ning Shen. 2025. Systematic discovery of single-cell protein networks in cancer with shusi. *bioRxiv*, pages 2025–04.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.
- Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Daisy Zhe Wang, Zhenhailong Wang, Cheng Qian, Robert Tang, Heng Ji, and 1 others. 2025. Multiagentbench: Evaluating the collaboration and competition of llm agents. In *Proceedings of the 63rd Annual Meeting of the Association for*

## A Benchmark Construction Details

This appendix section details the comprehensive construction procedure for our evaluation benchmark. We describe the process of extracting problems from LiveIdeaBench-v2, handling duplicate and low-quality extractions, assigning domains, and sampling the final 500 benchmark problems.

### A.1 Source Dataset

We utilize LiveIdeaBench-v2 as our source dataset. Each entry contains a generated scientific idea alongside metadata such as keywords and model information. The primary field leveraged for problem extraction is the free-form idea field. This field exhibits high heterogeneity: in certain entries, the underlying problem is explicitly marked with labels such as `Problem:`, whereas in others, it appears under headings like `Background:` or `Challenge:`, or is merely implied within the narrative prose.

The dataset encompasses over one thousand unique keywords, spanning a diverse spectrum of scientific and technical domains. This characteristics makes it highly suitable for constructing a benchmark that captures a wide array of problem formulations, rather than repeatedly sampling from a narrow set of common topics.

### A.2 Problem Extraction

We extract a single problem statement from each source sample. The extraction input comprises the original row index, the keyword field, and the corresponding idea text. The primary objective is to recover the core research problem, gap, limitation, challenge, or background motivation addressed by the proposed idea.

We employ DeepSeek-V4-Flash (DeepSeek-AI, 2026) via the DeepSeek OpenAI-compatible chat completion API. Utilizing JSON-mode decoding, we request a structured output containing the row index, the extracted problem, the source type, and the model’s confidence. The model is instructed to prioritize explicit fields, such as `Problem`, `Challenge`, `Issue`, `Gap`, `Need`, `Limitation`, `Motivation`, `Background`, or `Context`. In cases where no explicit field is present, the model infers the problem by analyzing segments of the text that describe what is missing, difficult, costly, risky, unresolved, inefficient, or limited.

To mitigate computational costs and avoid redundant API invocations, we first deduplicate identical idea strings. If multiple dataset entries share the exact same idea text, we invoke the extraction model only once and map all corresponding original row indices within a shared `row_indices` field. Following the extraction process, identical normalized problem statements are subsequently merged.

### Extraction prompts.

#### System prompt.

```
You are a scientific problem-extraction engine.

Your task is to recover the core research problem addressed
by each input idea.
The problem may be stated as a challenge, gap, limitation,
need, bottleneck,
open question, or background motivation.

Extraction rules:
1. Return valid JSON only. Do not include markdown, comments,
   or explanations.
2. Return one output object for each input item, preserving
   its row_index.
3. Prefer explicit problem-bearing fields, including Problem,
   Challenge, Issue,
   Gap, Need, Limitation, Motivation, Background, and Context.

4. If no explicit field is present, infer the problem from
   sentences describing
   what is missing, unresolved, costly, slow, risky,
   inefficient, difficult, or
   currently limited.
5. Do not treat the proposed method, system, or intervention
   as the problem
   unless the input contains no recoverable problem or
   background motivation.
6. Remove conversational artifacts such as "Sure", "Okay", "
   Final Answer",
   word-count notes, list markers, and formatting boilerplate.

7. Preserve the scientific meaning of the source text. Do not
   introduce facts,
   domains, claims, or causal explanations that are not
   supported by the input.
8. Write the problem as a concise standalone statement,
   normally one sentence
   and at most two sentences.
9. If the input only contains a proposal, infer the closest
   concise underlying
   problem and set confidence to "low".

Use the following output schema:
{
  "items": [
    {
      "row_index": 0,
      "problem": "Current ecodesign workflows often delay
sustainability integration by relying on post-hoc
environmental assessments.",
      "source": "explicit_problem",
      "confidence": "high"
    }
  ]
}
```

Allowed source values:

- `"explicit_problem"`: the problem is directly stated in a problem-like field.
- `"explicit_background"`: the problem is stated in background or motivation text.
- `"inferred"`: the problem is inferred from the surrounding idea narrative.

Allowed confidence values: `"high"`, `"medium"`, or `"low"`.

#### User prompt template.

```
{
  "instruction": "Extract core research problems. Return only
    a JSON items array.",
  "items": [
    {
      "row_index": 0,
      "keywords": ["ecodesign"],
      "idea": "<free-form LiveIdeaBench-v2 idea text>"
    }
  ]
}
```

### A.3 Deduplication

We perform deduplication at two stages. First, before calling the extraction model, we deduplicate exact normalized idea strings. This avoids repeated model calls for identical source samples. When several rows share the same idea, the extracted problem is associated with all corresponding row indices.

Second, after problem extraction, we normalize extracted problem strings by lowercasing, trimming whitespace, and collapsing repeated spaces. Problems with identical normalized text are merged into one benchmark candidate. Their keywords and row indices are combined. This produces a pool of unique problem candidates.

### A.4 Quality Filtering

We apply lightweight heuristic filters to remove clear extraction failures while avoiding overly aggressive filtering. Specifically, we remove candidates that are empty, contain assistant-style chatter, are generic placeholders, are extremely short or long, or begin with proposal-like method descriptions. For English problems, we retain candidates with approximately 6–50 words.

The filters are intentionally conservative. Many valid scientific problems are short, such as “Habitat fragmentation threatens biodiversity.” We therefore avoid removing short statements unless they are clearly uninformative or malformed.

### A.5 Domain Assignment

To balance the final benchmark, each candidate problem is assigned to one of 22 scientific domains. The target domains are:

Physics, Biology, Systems Engineering, Earth Science, Pharmacy, Finance, Accounting, Anthropology, Chemistry, Astronomy, Data Engineering, Political Science, Systems Science, Pedagogy, Agricultural Science, Medicine, Data Science, Statistics, Psychology, Sociology, Mathematics and Logic, and Economics.

Domain assignment is based on keyword and problem-text matching with manually defined do-

main rules. For example, terms related to quantum mechanics, optics, or superconductivity are assigned to Physics; terms related to machine learning, neural networks, or data mining are assigned to Data Science; and terms related to geology, climate, or oceanography are assigned to Earth Science.

This domain assignment is used only for benchmark balancing and analysis; it is not used as a target label for model evaluation.

### A.6 Sampling the 500-Problem Benchmark

The final benchmark comprises 500 unique problems. We utilize a fixed random seed to ensure the reproducibility of the sampling process. The sampling procedure is designed to achieve three primary objectives:

1. preserve problem quality,
2. maintain keyword diversity, and
3. balance the selected problems across target domains.

We allocate 20 problems specifically to computer science and AI-related topics. These samples are identified via keyword and problem-text indicators such as artificial intelligence, machine learning, deep learning, neural networks, large language models, computer vision, data science, software, and computational. The remaining 480 problems are then sampled from the broader candidate pool.

For the full benchmark, we perform sampling across the 22 domains as evenly as possible. Given the total of 500 problems, this strategy yields 16 domains containing 23 problems each and 6 domains containing 22 problems each. Throughout the sampling process, we prioritize candidates whose keywords have not yet been covered, while applying a lightweight lexical near-duplicate filter to prevent the selection of highly similar problem statements.

### A.7 Benchmark Composition

Figure 6 summarizes the overall composition of the final 500-problem benchmark. Specifically, Figure 6(a) illustrates the domain distribution following the sampling procedure. The benchmark encompasses a total of 23 domains, including a dedicated *Computer Science* subset. The majority of domains comprise 22 problems each, whereas four domains contain 21 problems, and *Computer Science* consists of exactly 20 problems. This distribution reflects our specific sampling constraint, which

deliberately reserves 20 problems for computer- and AI-related topics while ensuring the remainder of the benchmark maintains an approximate balance across broad scientific and social-science disciplines.

Figure 6(b) offers a qualitative visualization of the benchmark within a two-dimensional semantic space. To achieve this, we first represent each extracted problem utilizing TF-IDF features, subsequently applying dimensionality reduction followed by t-SNE. Each data point corresponds to an individual benchmark problem, with its designated domain denoted by distinct colors and marker shapes. This visualization demonstrates that the selected problems span a broad semantic region rather than collapsing into a narrow set of local clusters, thereby substantiating the intended diversity of the benchmark across various domains and problem types.

## B Training and Evaluation Details for Impact Reward Models

This appendix section provides supplementary details regarding the training and evaluation of the proposed scientific-idea reward models. We structure the discussion around four core aspects: experimental protocol, backbone and baseline selection, training implementation, and evaluation methodology.

### B.1 Additional Methodological Details

**Why pointwise reward modeling.** Our overarching goal is to support scientific idea generation pipelines in which a generator proposes multiple candidates and a judge must systematically assess their impact. A reusable pointwise reward is preferable to a pair-specific classifier because candidate ideas often need to be filtered, ranked, cached, rescored, and compared across diverse decoding runs. Accordingly, the judge is implemented as a scalar reward function over individual candidates, while pairwise preferences are induced on demand by score differences. Compared with prompt-conditioned generative judges or strictly pairwise classifiers, this interface more naturally supports sample reranking and future extensions involving reinforcement learning, search, or planning over generated scientific ideas.

**Structured input format.** We intentionally do not train on the benchmark’s original conversational prompts. Instead, each instance is converted

into the following unified representation:

```
[Field] {category}
[Subfield] {subcategory}
[Title] {title}
[Abstract] {abstract}
```

This design reduces prompt-specific surface variation and aligns reward-model training with the representation passed by our downstream generator-based system. Citation counts and ground-truth preference labels are excluded from the model input to avoid leakage.

**Backbone interface.** The framework supports both encoder-style and decoder-only backbones. Encoder-style models use a lightweight scalar reward head over the pooled sequence representation. Decoder-only models use the hidden state of the last non-padding token, followed by a scalar reward head, and are trained either by full fine-tuning or by parameter-efficient adaptation. This shared interface allows heterogeneous backbone families to produce comparable pointwise scores while still supporting pairwise comparison through score differences.

**Weighted pairwise training.** The weighting scheme in the main text is designed to emphasize pairs with clearer preference signals. Here  $\Delta_{\text{abs}}$  denotes the absolute citation gap and  $\Delta_{\text{rel}}$  the corresponding relative gap. The preliminary score  $\tilde{w}$  increases with both quantities, while the clipping operation constrains the final weight  $w$  to a bounded interval to avoid unstable gradients from unusually large citation differences. In our implementation, citation-derived statistics are used exclusively for weighting and offline analysis rather than as model inputs.

The weighted objective multiplies the standard Bradley–Terry loss by  $w$ , thereby assigning stronger optimization pressure to more distinguishable preference pairs. The regularization term  $\mathcal{L}_{\text{reg}}$  penalizes excessively large reward magnitudes through the coefficient  $\lambda$ , which helps stabilize optimization and mitigates uncontrolled score drift. In practice, weighted optimization is enabled only in selected experimental branches, and the epoch at which it begins is treated as a tunable schedule parameter.

**Training protocol overview.** We train on pairwise examples derived from the official training split, while reserving the benchmark evaluation

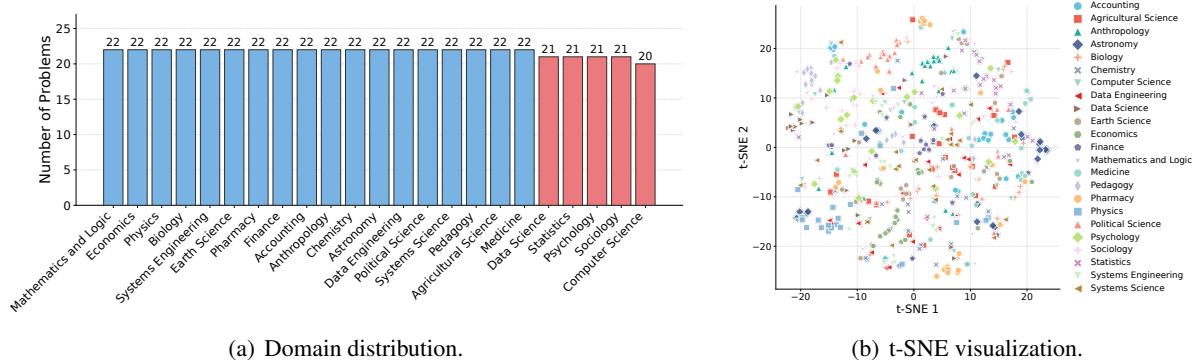


Figure 6: Composition of the 500-problem benchmark. The benchmark covers diverse scientific domains, and the t-SNE visualization is based on TF-IDF semantic representations, where colors and marker shapes denote assigned domains.

split for final testing. For large decoder-only backbones, training uses LoRA adapters (Hu et al., 2022), distributed data parallelism, mixed precision when supported, gradient accumulation, and pre-tokenized caches to reduce CPU-side preprocessing overhead. We also support staged schedules in which unweighted optimization precedes weighted fine-tuning, together with learning-rate warmup and either cosine or linear decay.

## B.2 Experimental Protocol

**Task formulation.** All reward models are trained to assign a scalar score to a candidate scientific idea. Each candidate is represented as a structured textual input, and pairwise supervision is provided via a preferred item  $x^+$  and a non-preferred item  $x^-$ . Although the training phase relies on pairwise preferences, the resulting model operates as a pointwise reward function during inference. Consequently, the same model can interchangeably score an individual candidate idea or induce a pairwise preference based on score differentials.

**Training and validation split.** We construct training pairs from the official training split of SciJudgeBench. In the current implementation, checkpoint selection follows the validation protocol instantiated in the training pipeline, while the benchmark evaluation split is retained for final reporting. Unless otherwise specified, the optimal checkpoint is selected based on pairwise accuracy under this validation protocol.

**Reported evaluation splits.** All reported experiments evaluate performance on the in-domain test split.

## B.3 Backbones and Reference Baselines

**Reward-model backbones.** We instantiate our reward-modeling framework on five backbone families: SciBERT-0.1B, SciTulu-7B, Galactica-6.7B, Qwen2.5-7B-Instruct, and Llama-3.1-8B-Instruct. We denote the resulting trained reward models as JudgeReward-SciBERT-0.1B, JudgeReward-SciTulu-7B, JudgeReward-Galactica-6.7B, JudgeReward-Qwen2.5-7B-Instruct, and JudgeReward-Llama-3.1-8B-Instruct, respectively.

**Scientific-domain models.** SciBERT serves as a lightweight, encoder-style scientific text baseline. It is highly efficient to train and establishes a stable reference point for scientific reward modeling. SciTulu-7B and Galactica-6.7B are incorporated as scientific-domain, decoder-only backbones. They are particularly relevant because their pretraining and instruction-tuning corpora contain strong scientific-literature priors, which may prove advantageous when evaluating the impact of generated research ideas.

**General-purpose instruction-tuned models.** Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct are included as robust, open-source general-purpose baselines. They enable us to investigate whether broad instruction-following capabilities transfer effectively to scientific idea evaluation, and to determine whether scientific-domain pretraining confers additional benefits beyond general language-modeling proficiency.

**SciJudgeBench-trained reference baselines.** In addition to our trained reward models, we compare against two SciJudgeBench-trained judge models:

SciJudge-Qwen2.5-32B and SciJudge-Qwen3-30B. These models act as benchmark-specific reference baselines. We evaluate them under the exact same reported split protocol whenever applicable, utilizing the test.

#### B.4 Reward Model Training

**Encoder-style reward model.** For SciBERT, we attach a lightweight scalar reward head to the pooled sequence representation. The encoder and the reward head are optimized jointly using the pairwise objective. This configuration requires neither adapter loading nor generation-specific infrastructure, rendering it comparatively simple, efficient, and stable.

**Decoder-only reward models.** For decoder-only backbones, we extract the hidden state of the last non-padding token to serve as the pooled representation. A scalar reward head is subsequently attached to this representation to output the final score. This architectural design ensures a unified scoring interface across all decoder-only models:

$$s_{\theta}(x) = \text{RewardHead}(\text{Pool}(\text{Backbone}(x))) \quad (19)$$

For Galactica-6.7B, we explicitly verify that the tokenizer possesses a valid padding token. Whenever necessary, token embeddings are resized prior to loading adapters or evaluating checkpoints.

**Parameter-efficient adaptation.** For large decoder-only models, we employ LoRA adapters in lieu of full fine-tuning. These adapters are integrated into the attention projections and, where applicable, the feed-forward projections. The specific target modules vary across backbone families due to differences in their underlying transformer implementations. The reward head itself remains fully trainable. This strategy substantially reduces the number of trainable parameters while preserving the scalar reward interface essential for the downstream generation pipeline.

**Training objective.** All reward models are trained utilizing the Bradley–Terry style pairwise logistic objective defined in Section 3.3. When enabled, we additionally incorporate bounded citation-gap-based weighting alongside a lightweight score-magnitude regularizer. The weighted objective prioritizes more distinguishable preference pairs, whereas the score regularization

helps stabilize the magnitude of the predicted rewards. In specific experimental runs, we adopt staged training schedules wherein unweighted optimization strictly precedes weighted fine-tuning.

**Optimization.** We support both linear and cosine learning rate schedules with configurable warmup periods. For large decoder-only backbones, we utilize small per-device batch sizes coupled with gradient accumulation to simulate larger effective batch sizes. Gradient norm clipping is applied to enhance training stability, particularly when weighted pairwise losses are active. Mixed-precision training is leveraged whenever supported by the model architecture and underlying hardware.

**Distributed training and evaluation.** Large decoder-only reward models are trained utilizing distributed data parallelism across multiple GPUs. The training data are sharded across ranks using distributed samplers, and gradient accumulation is employed to manage memory consumption. In the current implementation, validation and testing are likewise sharded across ranks, and predictions are gathered before final metric computation. This approach mitigates the bottlenecks of slow single-process evaluation and prevents non-main ranks from idling during extensive evaluation phases.

**Caching and preprocessing.** Raw benchmark records are initially converted into standardized pairwise examples, preserving both the preferred and non-preferred candidates alongside relevant metadata such as citation gaps and domain labels. For decoder-only models, we systematically pre-tokenize the training and evaluation pairs, storing them in model-specific caches keyed by backbone identity and maximum sequence length. This mechanism eliminates repeated tokenization across sequential runs and significantly reduces CPU-side preprocessing overhead during multi-GPU training and evaluation.

**Checkpointing and resumption.** Each checkpoint saves the adapted backbone, reward head, tokenizer, and, when applicable, the optimizer and scheduler states. For decoder-only models fine-tuned via LoRA, the saved backbone parameters consist exclusively of adapter weights rather than a fully materialized dense model. We support both full-state resumption and weight-only initialization. The latter proves particularly useful when resuming from a robust reward model checkpoint while in-

tentionally modifying the optimizer state, learning rate schedule, or stage-specific hyperparameters.

## B.5 Evaluation Protocol

**Evaluation splits.** All models are evaluated on the in-domain test split. We consistently apply these evaluation splits to both our trained reward models and the SciJudgeBench-trained reference baselines, wherever applicable.

**Pairwise accuracy.** Our primary evaluation metric is pairwise accuracy. Given an evaluation set comprising  $N$  preference pairs, each pair consists of a preferred item  $x_i^+$  and a non-preferred item  $x_i^-$ . A prediction is deemed correct if the reward model assigns a strictly higher score to the preferred item:

$$\text{Acc} = \frac{1}{N} \sum_{i=1}^N \mathbb{1} [s_{\theta}(x_i^+) > s_{\theta}(x_i^-)] \quad (20)$$

This metric directly reflects whether the learned reward model successfully recovers the underlying preference ordering established by the benchmark.

**Domain-wise accuracy.** In addition to the overall pairwise accuracy, we report domain-specific accuracies across CS, Math, Physics, and Others. This detailed breakdown serves to evaluate whether the performance of the reward model remains consistent across diverse scientific disciplines or is disproportionately concentrated within a specific subset of domains.

## B.6 Training Hyperparameter Settings

**Common settings.** Unless otherwise noted, all models are trained with the same pairwise logistic objective, a maximum input length of 512 tokens, gradient clipping, AdamW optimization, and the shared structured input format described in the main method section. For all large decoder-only models, we use distributed data parallel training, tokenized data caches, and a scalar reward head over the last non-padding token representation.

**SciBERT.** For the encoder-style SciBERT reward model, we use a batch size of 8, learning rates in the  $10^{-5}$  range, full-model optimization, and either linear or cosine learning-rate decay depending on the experimental branch. The validation split is held out from the benchmark training split, and the best checkpoint is selected according to validation pairwise accuracy.

**SciTulu-7B.** For SciTulu-7B, we use LoRA adapters with rank 8 or 16 depending on the branch, a per-device batch size of 1, and gradient accumulation of 8 steps. Typical learning rates range from  $2 \times 10^{-5}$  to  $10^{-4}$ , with cosine scheduling and warmup. Most stable runs disable gradient checkpointing and use bf16 mixed precision.

**Galactica-6.7B.** For Galactica-6.7B, we again use LoRA-based reward modeling with rank 8 in the main experimental branches. The standard 8-GPU setup uses a per-device batch size of 1, gradient accumulation of 8, cosine scheduling, and bf16 training in the more stable variants. We additionally control whether citation-gap weighting is enabled, the epoch at which weighting starts, and the clipping range of the pairwise weight term.

**Qwen2.5-7B-Instruct.** For Qwen2.5-7B-Instruct, we use LoRA rank 8, a per-device batch size of 1, gradient accumulation of 8, and warmup followed by scheduled decay. The backbone is trained in bf16, with LoRA applied to decoder projection layers and the reward head fully trainable.

**Llama-3.1-8B-Instruct.** For Llama-3.1-8B-Instruct, we use a training recipe matched to the other 7B–8B decoder-only baselines: LoRA rank 8, per-device batch size 1, gradient accumulation of 8, bf16 precision, and cosine or linear learning-rate scheduling depending on the experimental branch. This matched recipe allows more direct comparison against other large open-source instruction-tuned backbones.

**Backbone-specific LoRA targets.** The exact LoRA target modules depend on the transformer implementation. For LLaMA-style and Qwen2-style backbones, we adapt attention and feed-forward projections of the form q\_proj, k\_proj, v\_proj, o\_proj, gate\_proj, up\_proj, and down\_proj. For OPT-style backbones such as Galactica-6.7B, we adapt q\_proj, k\_proj, v\_proj, out\_proj, fc1, and fc2.

## C Compute Resources

For inference and evaluation, we use a workstation running Ubuntu 22.04 with 2 Intel Xeon Gold 6530 CPUs and an NVIDIA GeForce RTX 4090 GPU. For model training, we use 8 NVIDIA GeForce RTX 4090 GPUs with the same hardware specification as the inference GPU. Training is imple-

mented with PyTorch Distributed Data Parallel (DDP), where each GPU runs one process and processes a disjoint shard of the mini-batch. Gradients are synchronized across GPUs through NCCL all-reduce after back-propagation.

## D Qualitative cases.

Figures 7–9 show three cases where PRISM is ranked first by the novelty judge and also receives the highest impact score among the five methods. Figure 10 shows an additional case from the human expert evaluation where PRISM receives the maximum human score while all baselines receive 3. We show the generated ideas directly, with only minor compression of repeated evaluation boilerplate for space. The cases span healthcare access, environmental treatment, secure communication, and soil ecology. A separate inspection of the raw generations found that the PRISM ideas are more specific and operationally defined than the four displayed baselines in these examples; we use them as qualitative illustrations rather than additional quantitative evidence.

## E Inspection Views Used in PRISM

PRISM uses four fixed candidate inspection views, designed to cover complementary planning dimensions of a research problem. Table 6 lists each view with its identifier, name, and focus description. During generation, the entropy-based selector scores all four views against the task context and selects the top- $k$  (default  $k = 3$ ) to condition the backbone generator.

The views are provided to the backbone generator via the following prompt structure. The system prompt instructs the model to produce a rigorous, novelty-oriented research idea in JSON format. The user prompt supplies the task context and the selected views as planning constraints:

Listing 1: Generation prompt structure.

```
[System]
You generate rigorous, novelty-oriented but testable
scientific
research ideas and return valid JSON only.

[User]
Generate one strong research idea for this task.
Return only valid JSON. Do not use Markdown.

Task:
{task_context}

Selected views:
[
  {"view_id": "V1", "name": "...", "focus": "..."},
  {"view_id": "V3", "name": "...", "focus": "..."},
  ...
]
```

```
]

Instructions:
- Use the selected views as planning constraints for the idea.

- Do not list the views as separate sections.
- Integrate their substance into a single coherent proposal
  that
  explains the core mechanism, expected advantage,
  feasibility,
  and a minimal validation plan.
```

## F Domain Robustness

Figure 11 reports a domain-level breakdown of the pairwise win rates. For each domain, we compare PRISM with the four baselines over all backbone-problem groups assigned to that domain. The impact win rate ranges from 72.1% in Chemistry to 88.0% in Anthropology, and the novelty win rate ranges from 84.5% in Statistics to 95.7% in Chemistry. These results suggest that the aggregate gains are not concentrated in only one or two domains. The domain variation is also informative: impact varies more than novelty, which is expected because the impact judge is sensitive to whether a proposal looks useful under domain-specific research norms, while the novelty judge mainly compares originality among the five generated ideas for the same problem.

## Case 1: Federated Rare-Disease Counseling

[Backbone] Qwen3-8B

[Problem] Access to genetic counseling is often limited, especially for underserved populations and individuals with rare diseases.

[Judge Outcome] PRISM novelty rank 1; impact score 0.10. Strongest baseline impact: VirSci -2.29.

### [Generated Ideas]

**PRISM.** We propose a decentralized, AI-powered genetic counseling platform that combines natural language processing with federated learning to provide personalized rare-disease risk assessments for underserved populations. The system fine-tunes low-resource language models on rare-disease data while keeping sensitive genetic records local, then improves through user feedback and oversight from local healthcare providers. Its main advantage is a privacy-preserving and culturally adaptive counseling workflow for communities with limited specialist access. Validation would use a pilot across underserved regions, measuring satisfaction, risk-prediction accuracy, and specialist referrals.

**ResearchAgent.** This research proposes a centralized rare disease genetic database integrated with natural language processing to dynamically match patients with specialized genetic counselors. The system aggregates anonymized genetic data and clinical histories, uses NLP to identify patient-specific risk factors, and recommends counselors with relevant expertise. A hybrid telehealth framework enables secure virtual consultations, while culturally adaptive interfaces improve engagement. Feasibility relies on existing NLP tools, telehealth infrastructure, and genetic counseling networks. Validation measures match success, diagnostic concordance, and rural-urban health-equity metrics.

**AI Scientist.** This research proposes an AI-augmented telehealth platform combined with mobile-first training modules for community health workers. The platform uses AI risk assessment in telehealth interfaces to prioritize high-risk cases for specialist intervention, while mobile modules help community workers provide basic genetic risk assessment and triage guidance. Feasibility is based on existing telehealth infrastructure and modular AI frameworks. A validation experiment would compare AI-assisted care with standard care in rural clinics.

**AI Researcher.** This research proposes an AI-powered virtual genetic counseling platform integrated with telehealth infrastructure. The system analyzes patient-reported family histories and genetic test results, generates personalized risk assessments, and flags complex cases for human review. The dual-layer design balances automated triage with specialist oversight. Feasibility is supported by telehealth adoption and modular AI components. Validation focuses on clinical accuracy and patient-reported outcomes for virtual counseling quality.

**VirSci.** This research proposes an AI-augmented telehealth platform to prioritize and deliver genetic counseling for underserved rare-disease populations. NLP algorithms analyze patient histories and identify high-risk cases, while telehealth infrastructure connects patients with certified counselors in real time. The approach combines AI-driven triage with remote service delivery, with risks from cross-state regulation and potential bias in rare-disease risk scoring. Validation compares AI-assisted triage with traditional rural-clinic workflows.

Figure 7: Generated ideas for a rare-disease counseling problem.

## Case 2: Acid-Rain Pollutant Valorization

[Backbone] Llama-3.1-8B

[Problem] Conventional acid rain treatments neutralize acidity but do not recover valuable compounds or create economic value from pollutants.

[Judge Outcome] PRISM novelty rank 1; impact score 0.33. Strongest baseline impact: AI Researcher -2.40.

### [Generated Ideas]

**PRISM.** We propose a hybrid acid-rain treatment that combines microbial electrolysis with biochar-based carbon capture. Electroactive bacteria break down pollutants into valuable chemicals, while biochar captures and stabilizes the recovered products for downstream processing. The goal is to recover economic value from pollutants while reducing acid-rain damage. The main risk is contamination or drift in the microbial community, so the system includes monitoring of microbial health and diversity. Validation would test recovery yield, pollutant reduction, and system stability in a controlled field setting.

**ResearchAgent.** Developing a closed-loop system for acid rain treatment and compound recovery addresses the inefficiency of neutralization-only methods. The idea proposes using microorganisms or chemical processes to break down pollutants and recover valuable compounds, creating a system that mitigates acid-rain damage and generates economic value. The proposal notes risks from pilot cost, market uncertainty, and environmental safety. Validation would test whether valuable compounds can be recovered from acid-rain pollutants in a pilot study.

**AI Scientist.** This idea develops a membrane-based technology that separates and recovers valuable compounds from acid rain. The membrane system would selectively capture and convert pollutants into useful chemicals such as fertilizers, fuels, or other commodities. The proposal frames the method as a closed-loop treatment system, while noting scale-up challenges, environmental impacts, and regulatory hurdles. Validation would test membrane efficacy in a controlled pilot setting.

**AI Researcher.** Developing an electrochemical process to recover valuable compounds from acid-rain pollutants would neutralize acid rain and extract useful compounds in the same treatment pipeline. The approach aims to create a closed-loop system that reduces waste and increases resource utilization. Risks include high energy requirements and variability in pollutant composition. Validation would combine literature review, expert consultation, and pilot-scale protocol design for an electrochemical recovery process.

**VirSci.** This idea develops a nanotechnology-based system to recover valuable compounds from acid-rain pollutants. Nanomaterials are used to improve pollutant recovery, and electrochemical processes convert pollutants into valuable byproducts. The proposal highlights possible environmental benefits but also notes high energy demand and toxicity risks. Validation would use literature review, expert consultation, economic analysis, and computational simulation of the proposed recovery system.

Figure 8: Generated ideas for an acid-rain treatment problem.

### Case 3: Policy-Bound Secure Communication

[Backbone] GPT-5.4

[Problem] There is a growing need for secure communication in sensitive sectors like finance and defense.

[Judge Outcome] PRISM novelty rank 1; impact score 0.18. Strongest baseline impact: ResearchAgent -1.38.

#### [Generated Ideas]

**PRISM.** We propose LedgerShield, a policy-bound dual-channel secure messaging system for finance and defense workflows. Instead of only strengthening encryption during transport, LedgerShield targets semantic containment after decryption, where leaks can occur through forwarding, context mixing, model-assisted summarization, or insider misuse. Each message carries a machine-verifiable policy capsule that encodes permitted recipients, allowed transformations, expiration, sensitivity class, and audit obligations. Trusted endpoint execution enforces compliant reading, quoting, and summarization, while cryptographic receipts are written to an append-only compliance ledger. Validation would compare LedgerShield with encrypted messaging on seeded insider and accidental leakage tasks, measuring unauthorized disclosure, workflow completion, latency, and audit completeness.

**ResearchAgent.** We propose a deployable secure communication architecture for latency-sensitive financial operations, with defense networks as a secondary setting. The core mechanism is a policy-driven hybrid channel that combines classical key exchange with post-quantum KEMs, binds session establishment to endpoint attestation when available, and triggers adaptive rekeying or failover when device trust, network conditions, or communication criticality change. Validation compares classical-only, static hybrid, and adaptive hybrid designs on latency, bandwidth overhead, rekey cost, and compromise-recovery delay.

**AI Scientist.** We propose a payload-free detection framework for suspicious secure communications in finance-grade environments. The method fuses temporal flow statistics, evolving communication graphs, endpoint role and user context, and protocol or session fingerprints. A lightweight multi-view model is trained with scarce incident labels and semi-synthetic threat injections such as beaconing, covert peer-group formation, abnormal privileged messaging, and unusual fan-out from sensitive endpoints. Validation reports AUROC, AUPRC, and top-k alert precision across injected attack types.

**AI Researcher.** We propose a policy-aware secure messaging protocol for cross-institution communication in finance and defense that unifies post-quantum key establishment, forward secrecy, and organization-level access control. A hybrid classical/post-quantum handshake derives ephemeral session keys while binding each session to signed machine-readable policies for recipient attributes, retention limits, forwarding restrictions, and approval rules. Validation prototypes the handshake and policy-check pipeline and compares latency, failure modes, and security properties against enterprise end-to-end messaging.

**VirSci.** We propose an auditable secure communication fabric for finance and defense operations that integrates hybrid classical/post-quantum key exchange, hardware-backed endpoint attestation, continuous session rekeying, and policy-controlled metadata minimization. Sessions are established after endpoint trust checks, key derivation, and binding to organization-defined communication policies; high-risk flows can activate timing, batching, and route-diversity controls. Validation compares the fabric with standard encrypted channels on metadata leakage, rekeying overhead, attestation delay, and audit compatibility.

Figure 9: Generated ideas for a secure communication problem.

### Case 4: Microplastic Mechanisms in Soil Invertebrates

[Backbone] GPT-5.4 [Domain] Agricultural Science

[Problem] The impact of microplastic pollution on soil ecosystems, particularly its effects on subterranean invertebrates, is poorly understood.

[Human Scores] PRISM=5; AI Researcher=3; AI Scientist=3; ResearchAgent=3; VirSci=3

#### [Generated Ideas]

**PRISM.** We propose RhizoLoom, a trait-resolved soil mesocosm method that links isotopically labeled microplastics, species-specific invertebrate movement imaging, and microbial exometabolite profiling to identify whether microplastics harm subterranean invertebrates through physical habitat disruption, chemical leachate stress, or trophic re-routing. The novelty gap is that current soil microplastic studies usually report bulk toxicity or abundance shifts, which cannot reject false-positive explanations such as moisture loss, food dilution, or generic particle irritation. RhizoLoom addresses this by introducing paired polymer-specific and geometry-matched inert controls, plus root-present and root-absent soil compartments, so the signal is attributed to a specific mechanism rather than a generic particle effect.

**ResearchAgent.** We propose a function-linked soil ecotoxicology study testing whether environmentally realistic, weathered microplastic fibers and fragments disrupt subterranean invertebrates and the ecosystem processes they drive in agricultural soils. The core mechanism is that particle shape and aging state alter ingestion, cuticle and burrow contact, and soil pore structure, thereby changing earthworm and collembolan behavior, oxidative-stress responses, and feeding activity, with downstream effects on litter decomposition, aggregate stability, and mineral nitrogen release.

**AI Scientist.** This project proposes a controlled soil mesocosm study to test whether microplastic aging changes how subterranean invertebrates mediate soil ecosystem functioning, using earthworms as a tractable engineer species. The central mechanism is that pristine versus environmentally aged microplastics differ in surface chemistry and particle interactions, which may alter ingestion, burrowing, casting, and stress responses in earthworms, with downstream effects on aggregate stability and litter decomposition.

**AI Researcher.** This project proposes a soil mesocosm study to test whether realistic, weathered agricultural microplastic mixtures affect subterranean invertebrates primarily by altering habitat structure and moisture dynamics, rather than through direct toxicity alone. The core mechanism is that irregular, aged particles can change pore connectivity, aggregation, and water retention, which then shifts microbial activity and food availability for taxa such as nematodes, collembolans, enchytraeids, and juvenile earthworms.

**VirSci.** We propose a mechanism-centered mesocosm study testing whether microplastics harm subterranean soil invertebrates primarily by reorganizing soil physical microhabitats rather than only through direct toxicity. The core hypothesis is that agriculturally realistic inputs of microplastics alter pore connectivity, water retention, and microscale organic matter distribution, which then changes the movement, burrowing, feeding, and reproduction of two contrasting belowground guilds.

Figure 10: Generated ideas for a soil microplastic problem (human expert evaluation). PRISM receives the maximum human score (5) while all four baselines receive 3. RhizoLoom’s three-pronged mechanistic design—*isotope labeling, movement imaging, and exometabolite profiling*—provides a concrete path to distinguish causal mechanisms, whereas the baselines converge on similar mesocosm designs without a mechanism-attribution strategy.

| ID | Name                                      | Focus                                                                                                             |
|----|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| V1 | Novelty Gap with False-Positive Rejection | Frame the idea around an unmet gap and explicitly reject false-positive, trivial, or already-solved explanations. |
| V2 | Mechanism or Intervention                 | Specify the concrete mechanism, algorithm, intervention, or scientific move that solves the gap.                  |
| V3 | Risk and Uncertainty Calibration          | Identify uncertainty, confounds, failure modes, and how the idea calibrates or controls them.                     |
| V4 | Minimal Validation Benchmark              | Design the smallest convincing experiment, benchmark, or prospective test with measurable success criteria.       |

Table 6: The four candidate inspection views used in PRISM. For each task, the entropy-based selector scores all views and selects the top- $k$  most informative ones to condition the generator.

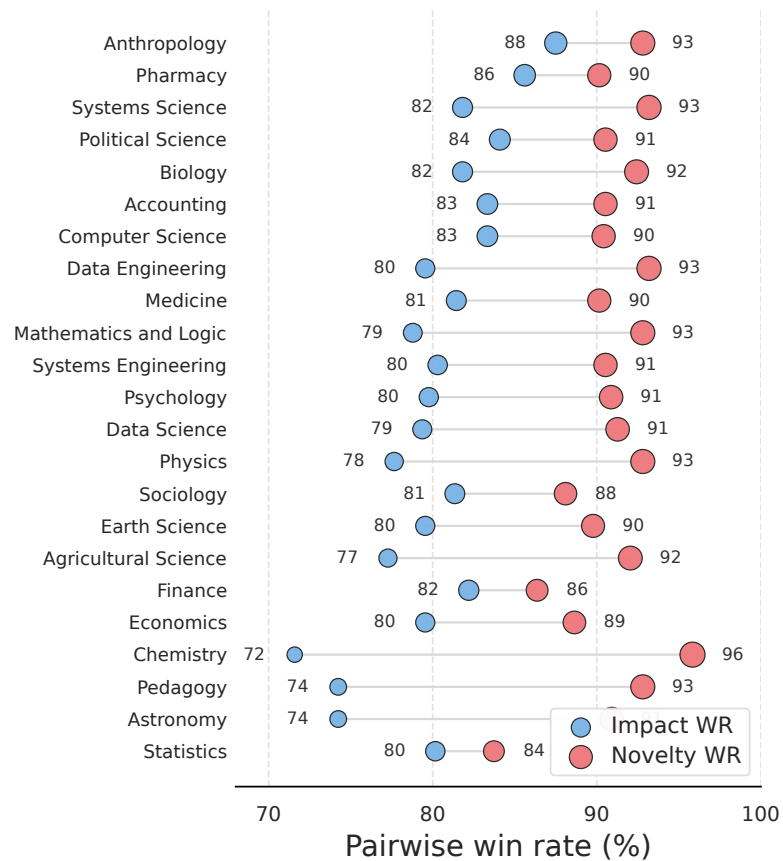


Figure 11: Domain-level PRISM robustness. Each point reports the pairwise win rate against the four baselines over backbone-problem groups in that domain. Gray line segments connect the impact and novelty rates for the same domain.