FLOWPIE: TEST-TIME SCIENTIFIC IDEA EVOLUTION WITH FLOW-GUIDED LITERATURE EXPLORATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Scientific idea generation (SIG) is central to AI-driven autonomous research, yet existing approaches often rely on a static retrieval-then-generation paradigm, which limits literature exploration and leads to homogeneous, insufficiently divergent ideas. In this work, we propose FlowPIE, a tightly coupled retrieval–generation framework that treats literature exploration and idea generation as a co-evolving process. FlowPIE first constructs a diverse and high-quality initial idea population through flow-guided Monte Carlo Tree Search (MCTS), inspired by GFlowNets, where the quality of generated ideas, assessed by an LLM-based generative reward model (GRM), serves as feedback to guide adaptive literature exploration. FlowPIE then formulates idea generation as test-time idea evolution, applying selection, crossover, and mutation with an isolation island paradigm and GRM-based fitness evaluation to incorporate cross-domain knowledge and refine ideas beyond the initial retrieval space. This design mitigates the information cocoon effect caused by over-reliance on parametric knowledge and static literature. Extensive evaluations on AI Idea Bench 2025 and IdeaBench show that FlowPIE consistently generates ideas with higher novelty, feasibility, relevance, and diversity than strong LLM-based and agent-based baselines. Notably, even its initial idea population outperforms several competing methods, while subsequent evolution yields clear test-time reward scaling, robust convergence, and strong domain generalization. Human evaluation further confirms that FlowPIE produces higher-quality ideas aligned with expert preferences. Our code and datasets will be released to facilitate future research on autonomous scientific ideation.

1 INTRODUCTION

With the rapid development of large language models (LLMs) (Hurst et al., 2024; Liu et al., 2024), their strong multidisciplinary understanding and reasoning capabilities have made it increasingly feasible to synthesize knowledge from large-scale scientific literature. Recent efforts have shown that LLM-based and agent-based systems can support the entire scientific research pipeline, from proposal development and experimental design to result analysis and paper drafting, shaping an emerging paradigm of autonomous research (Chen et al., 2025).

Scientific idea generation (SIG) has emerged as a key frontier in autonomous research, attracting significant attention across diverse domains. As shown in Figure 1, most existing methods mine novel ideas from literature databases using a decoupled two-stage framework: first *retrieving relevant literature*, and then *generating ideas based on the retrieved literature*. This pipeline typically relies on a single retrieval step driven by keyword matching and semantic relevance (Wang et al., 2024) to a specific topic. However, using this static retrieval step as the primary source of inspiration often yields contexts that are only topically similar, rather than genuinely conducive to innovation. As a result, the retrieved knowledge remains limited in both depth and breadth, frequently leading to homogeneous ideas with limited divergence.

In the idea generation stage, prior works leverage LLM-based brainstorming (Wang et al., 2024), research agents with review (Baek et al., 2025), or multi-agent discussion (Su et al., 2025) to generate and refine ideas. These approaches exploit the parametric knowledge encoded in LLMs together with information from statically retrieved literature. However, such designs risk trapping LLM-based generators within an information cocoon, bounded by their internal knowledge and a fixed

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

set of external sources. These limitations motivate us to revisit the widely adopted retrieval-and-generation paradigm for SIG. Specifically, we focus on the following two research questions (RQs): ***RQ1***: How can literature retrieval become a *dynamic*, *adaptive* component of idea generation, rather than a *static* preliminary stage? ***RQ2***: How can LLMs leverage retrieved literature and their relationships to generate novel and divergent ideas through continuous refinement?

In this work, we propose **FlowPIE**, a tightly coupled retrieval–generation framework for test-time idea evolution, as illustrated in Figure 3. Moving beyond the traditional static retrieval paradigm, FlowPIE unifies literature retrieval and idea generation into a dynamic and adaptive ***test-time idea evolution process***. Within this process, intermediate generated ideas serve as active feedback to guide subsequent literature exploration. Specifically, FlowPIE performs structured exploration over the literature subgraph by modeling retrieval as flow-guided MCTS inspired by GFlowNets (Bengio et al., 2023), thereby incrementally expanding retrieval trajectories in both breadth and depth.

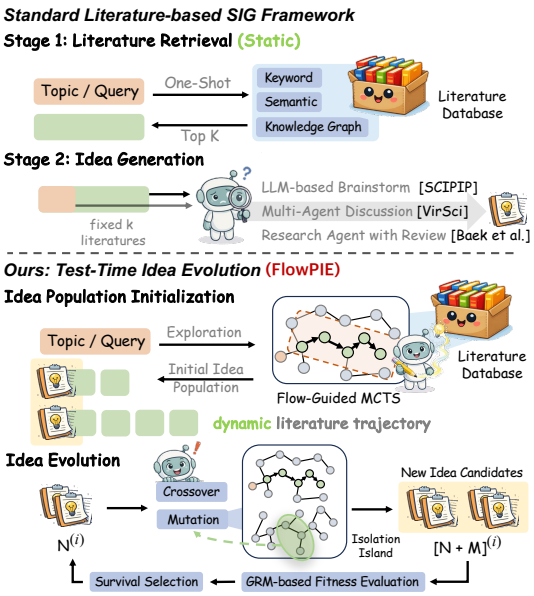

Figure 1: Comparison of traditional literature-based SIG frameworks and our FlowPIE.

The ideas generated along these paths are then organized into an initial population for subsequent evolution. FlowPIE is implemented as an evolutionary algorithm that operates over this idea population through the iterative application of selection, crossover, and mutation operators, with fitness evaluated by an LLM-based generative reward model (GRM). During mutation, it further introduces the *isolation island* paradigm, which maintains multiple isolated literature regions to facilitate the incorporation of cross-domain knowledge and diverse characteristics.

Extensive experiments and human evaluations on AI Idea Bench 2025 (Qiu et al., 2025) and IdeaBench (Guo et al., 2025) demonstrate that FlowPIE outperforms strong LLM-based and agent-based baselines, generating ideas that are more novel, divergent, and generalizable across domains. Beyond benchmark performance, we analyze the reward scaling behavior of FlowPIE in Figure 2. The reward initially fluctuates during literature exploration, then increases as flow-guided search balances exploration and exploitation. After the initial ideas are obtained, continuous idea evolution further refines them toward higher-quality regions with more stable convergence. Notably, both the evolved ideas and the initial ideas achieve higher reward scores than those of competing baselines.
Our main contributions are as follows:

- We introduce **FlowPIE**, a tightly coupled retrieval–generation framework that reformulates scientific idea generation as a **test-time idea evolution** process. Unlike one-shot retrieval-and-generation pipelines, FlowPIE continuously refines an idea population through survival selection, crossover, and mutation under GRM-based fitness evaluation.
- We propose a **flow-guided MCTS** mechanism for adaptive literature exploration, where the quality of generated ideas provides feedback to update trajectory probabilities. This creates a closed-loop interaction between retrieval and generation, enabling the system to explore literature in both breadth and depth rather than relying on a fixed retrieved context.
- We design an **isolation-island-enhanced mutation** strategy that introduces topologically distant literature into the evolution process, promoting cross-domain knowledge transfer and mitigating the information cocoon effect caused by static LLM parametric knowledge.
- Extensive experiments on AI Idea Bench 2025 and IdeaBench, together with human evaluation, show that FlowPIE consistently improves idea quality, diversity, and domain generalization over strong LLM-based and agent-based baselines. Further reward-curve analysis demonstrates clear test-time reward scaling, with both the initial and evolved idea populations outperforming competing methods.

2 Related Work

AI for Science Research. With the advancement of LLMs, the landscape of scientific inquiry has been fundamentally reshaped across a wide range of disciplines, including physics (Ye et al., 2025), medicine (Liao et al., 2024), and mathematics (Romera-Paredes et al., 2023). Chen et al. (2025) established a holistic framework for AI for Research that systemizes current AI applications and explores the future trajectory of AI's impact on the research ecosystem. AI-Scientist (Lu et al., 2024) and AI-Researcher (Tang et al., 2025) aim to support the complete lifecycle of autonomous research, with the ultimate goal of producing a full research paper, including code generation and experimental execution. Co-Scientist (Gottweis et al., 2026) employs a multi-agent framework to iteratively generate, evaluate, and refine scientific hypotheses, highlighting the potential of agentic systems to improve the quality of research ideas through iterative evolution and scientific reasoning.

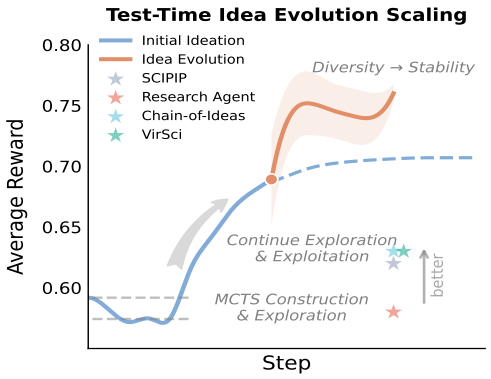

Figure 2: Test-time idea evolution scaling for reward. **Initial Ideation** couples literature exploration with idea generation via flow-guided MCTS, where higher reward, reflecting better initial ideas, amplify the weight of corresponding literature. **Idea Evolution** uses various evolutionary operators to guide ideas toward regions of higher rewards and stable convergence.

Scientific Idea Generation. Most previous SIG algorithms are rooted in simulations of the human ideation process. SCIPIP (Wang et al., 2024) leverages keywords and semantic similarity to retrieve relevant literature and synthesizes new ideas through LLM-based brainstorming. Li et al. (2025) consider relationships among literature and leverage a CoI agent to construct a chain of ideas before idea generation, using it as a curated future-direction prompt for subsequent synthesis. Baek et al. (2025) construct an entity-centric knowledge graph for literature survey, then generate and review ideas using a research agent and a review agent. Additionally, VirSci (Su et al., 2025) adopts a multi-agent system, including team construction and discussion, to simulate collaborative ideation.

We concur with these literature-based methods that new ideas arise from prior art rather than in a vacuum, and further argue that the quality of generated ideas is strongly constrained by the quality, diversity, and adaptiveness of the supporting literature. However, most existing methods retrieve literature in a static or weakly adaptive manner and then rely on LLMs or agents to generate ideas from a fixed context. In contrast, FlowPIE approaches idea generation from a test-time idea evolution perspective based on evolutionary algorithms (EAs), coupling literature retrieval with feedback from initial idea quality to enable adaptive literature exploration and continuous idea refinement. More discussion of related work on SIG evaluation and LLM-enhanced EAs is provided in Appendix A.

3 Methodology

In this work, we argue that scientific idea generation is not an isolated process. Rather, novelty must be grounded in technical realism and emerge cumulatively from the synthesis of prior scientific knowledge. Based on this perspective, we propose **FlowPIE**, as illustrated in Fig. 3, Algorithms 1 and 2. FlowPIE models the SIG process using an evolutionary algorithm. Specifically, the initial idea population is generated through dynamic literature exploration driven by flow-guided Monte Carlo Tree Search (MCTS) (see Sec. 3.1). This population is then refined through iterative evolution incorporating fitness evaluation, survival selection, crossover, and mutation operators(see Sec. 3.2).

Task and Idea Formulation. The goal of scientific idea generation is to formulate novel and diverse research hypotheses that can accelerate automated scientific discovery. Given a topic or query q, the idea generator aims to produce a set of structured scientific ideas grounded in existing knowledge and literature. Formally, the framework leverages an LLM-based idea generator to map q to an idea set $\mathcal{I} = \{I_1, ...I_k\}$, where each idea I is represented as a structured tuple comprising ***Motivation***, ***Method***, and ***Experimental Plan***. Unlike prior work such as SCIPIP, which primarily focuses on

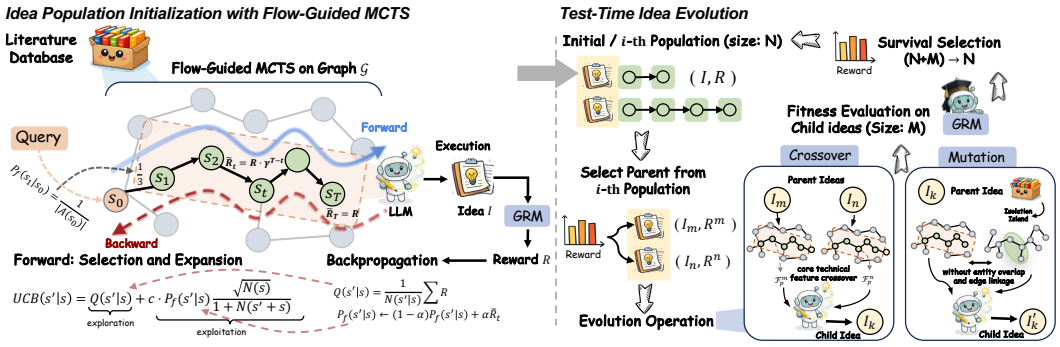

Figure 3: Overview of FlowPIE. **Left:** Idea initialization based on flow-guided MCTS: forward exploration via flow-guided UCB (Eq.1) for nodes selection and expansion, and backward updating (Eq.2) to enforce local and global flow constraints; this dual process dynamically adjusts literature weights with GRM-based rewards on generated ideas. **Right:** Test-time idea evolution, supervised by GRM-based fitness computation, leveraging various operators, including crossover on core technical features and isolation-island-enhanced mutation, to evolve ideas.

the problem–method structure, our formulation explicitly incorporates a detailed experimental plan following Baek et al. (2025), making the generated ideas more actionable and better aligned with practical research workflows.

3.1 Initial Idea Population Construction

Literature Graph Construction. To model relationships among literature, we construct hierarchical structured attributes for each document. Consider a literature database $\mathcal{D}$ spanning diverse research domains. Given any literature item $p \in \mathcal{D}$, we leverage an LLM to map p into an attribute tuple $\langle \mathcal{A}_p, \mathcal{F}_p, \mathcal{S}_p \rangle$, where $\mathcal{A}_p$ represents its abstract, $\mathcal{F}_p$ represents its core technical feature set extracted by the LLM, and $\mathcal{S}_p \in \mathbb{R}^d$ denotes its semantic embedding.

We formalize the literature graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the node set of unique literature items in $\mathcal{D}$. The edge set $\mathcal{E}$ represents relationships between nodes, where an edge $e_{ij} \in \mathcal{E}$ exists if p_i and p_j satisfy at least one of the following criteria: ***(i)*** a direct citation relation; ***(ii)*** overlap in core technical features, *i.e.*, $\mathcal{F}_i \cap \mathcal{F}_j \neq \varnothing$; ***(iii)*** semantic similarity $sim(\mathcal{S}_i, \mathcal{S}_j)$ exceeding a threshold. Further details on the literature corpus and graph construction are provided in Appendix B.2.

Idea Initialization with Literature Exploration via Flow-Guided MCTS. To construct a high-quality and diverse initial population of ideas, inspired by GFlowNet (Bengio et al., 2023), we propose a literature exploration mechanism, termed flow-guided MCTS, over graph $\mathcal{G}$. Given a query, we treat it as the root node s_0 and set its initial flow $F(s_0) = 1$, then retrieve relevant literature using similarity. For any node s with expandable adjacent node $s' \in A(s)$, the flow probability P_f is uniformly initialized as $P_f(s' \mid s) = \frac{1}{|A(s)|}$.

Selection and Expansion: We balance exploration and exploitation when selecting and expanding adjacent nodes along the edges of the constructed literature graph $\mathcal{G}$ by using a flow-guided Upper Confidence Bound (UCB):

$$UCB(s' \mid s) = Q(s' \mid s) + c \cdot P_f(s' \mid s) \frac{\sqrt{N(s)}}{1 + N(s' \mid s)} \tag{1}$$

where $Q(s' \mid s)$ denotes the expected value, encouraging exploitation of paths that have previously generated high reward ideas. $N(\cdot)$ denotes visit counts, and c is the exploration rate.

Execution and Backpropagation: The LLM-based idea generator produces an idea based on the currently explored literature trajectory, which then receives a reward R from the GRM. We backpropagate this reward to update the UCB value of trajectory in Equation 1. Considering the importance of literature at different depths, we introduce a depth-decayed reward $\tilde{R}_t = R \cdot \gamma^{T-t}$,

where T is the maximum depth of current trajectory. The value estimation is updated via standard averaging $Q(s' \mid s) = \frac{1}{N(s'|s)} \sum R^i$, while the flow probability is updated using a moving average:

$$P_f(s' \mid s) \leftarrow (1-\alpha)P_f(s' \mid s) + \alpha \tilde{R}_t \tag{2}$$

where $\alpha \in [0, 1]$ controls the weight of the reward.

Following this, P_f is locally normalized over $A(s_t)$ at each step, while being constrained by the global flow F, $P_f(s' \mid s) = \frac{F(s'|s)}{F(s)}$. Accordingly, the global flow is propagated forward as $F(s_{i+1}) = F(s_i) \cdot P_f(s_{i+1} \mid s_i)$. Exploration terminates when the reward variance of the generated ideas falls below a threshold ϵ. The resulting ideas, together with the literature traced from the root node s_0, are then used to initialize the idea evolution phase. Further details are provided in Algorithm 1 in the Appendix.

3.2 Test-Time Idea Evolution

The initial idea population constructed by flow-guided MCTS serves as an intermediate signal for broad and deep literature exploration. However, it lacks sufficient continuous refinement to further improve novelty and feasibility, leading to the reward bottleneck shown in Figure 2. In this section, we introduce test-time idea evolution within FlowPIE, which iteratively applies survival selection, crossover, and mutation operators to the initial population using an LLM-based idea generator.

Idea Evolution with Crossover and Mutation. We leverage an LLM-based idea generator to produce offspring ideas through pairwise crossover and isolation-island-enhanced mutation.

Crossover Operator. The crossover operator aims to synthesize advantages from different promising ideas. Given two parent ideas $\mathcal{I}_m$ and $\mathcal{I}_n$, along with the explored literature, we define the crossover process as $\mathcal{I}_k = \text{LLM}(q, \mathcal{I}_m, \mathcal{I}_n)$, where $\mathcal{I}_k$ denotes the generated offspring idea. Rather than performing superficial textual interpolation, this operator recombines the core technical features of the parent ideas under the guidance of the retrieved literature, enabling the LLM to synthesize a novel descendant idea that integrates their complementary characteristics and inherits their strengths.

Mutation Operator with Isolation Island. To prevent idea evolution from being trapped in local optima while maintaining diversity, we introduce a mutation operator governed by a mutation rate ρ. For each offspring $\mathcal{I}_k$, mutation is triggered by sampling $z_k \sim \text{Bernoulli}(\rho)$. If $z_k = 1$, we apply the ***Isolation Island*** strategy on graph $\mathcal{G}$. Instead of retrieving literature only from the current local neighborhood, we sample an auxiliary set $\mathcal{Q}_k^{\text{iso}} \subset \mathcal{G}$ from topologically distant subgraphs disconnected from the current neighborhood. The mutated idea is then generated by $\mathcal{I}'_k = \text{LLM}(q, \mathcal{I}_k, \mathcal{Q}_k^{\text{iso}})$, where the LLM is encouraged to logically integrate out-of-domain (OOD) information into $\mathcal{I}_k$, thereby expanding the conceptual boundaries of the idea.

Idea Fitness Evaluation. The offspring ideas obtained through evolution are then evaluated using the GRM. Specifically, the GRM assesses each idea across multiple dimensions, such as novelty and feasibility, which are subsequently aggregated into a scalar fitness score. Reward definitions and prompt details are provided in Appx. D.2 and F.

Survival Selection. We adopt a tournament selection strategy to form the next-generation population $\mathcal{I}_{next}$. Specifically, offspring and parent ideas are merged into a candidate pool C. While $|\mathcal{I}_{next}| < N$, we randomly sample a subset $S \subset C$, select the highest-reward idea, and add it to $\mathcal{I}_{next}$. This process repeats until $|\mathcal{I}_{next}| = N$, thereby preserving high-fitness ideas for the next generation. The evolution process stops when the maximum number of iterations is reached or the reward converges, yielding the final N evolved ideas.

4 Experiments

4.1 Experimental Setup

Literature database. We construct literature databases from two complementary sources: patent documents and scientific publications. For the main experiments, we use a patent corpus collected from publicly available USPTO resources, containing 16,050 patents published between October

Table 1: Main results on AI-Idea-Bench-2025. We generate three ideas with each method and report using the *mean* ± *std* for I2T and I2I. **Bold** indicates the best score, and underlining denotes the second-best score.

Method	I2T		I2I		IMCQ	
	Motivation	Exp Plan	Motivation	Exp Plan	Motivation	Exp Plan
SCIPIP	4.18±0.662	—	3.68±0.383	—	0.464	—
Research Agent	4.56±0.679	3.59±0.670	3.78±0.491	3.45±0.619	0.510	0.497
Chain-of-Ideas	4.63±0.703	3.79±0.566	3.74±0.504	3.50±0.653	0.507	0.433
VirSci	4.32±0.652	3.68±0.565	3.95±0.410	3.84±0.410	0.560	0.580
FlowPIE (*Ours*) w/ paper	**4.66±0.599**	**3.80±0.448**	4.31±0.322	3.80±0.461	0.755	**0.643**
Initial Population	4.52±0.516	3.75±0.616	4.20±0.322	3.70±0.461	0.657	0.626
FlowPIE (*Ours*) w/ patent	4.64±0.606	3.72±0.481	**4.44±0.318**	**3.85±0.347**	**0.780**	0.635
Initial Population	4.58±0.616	3.58±0.512	4.26±0.224	3.61±0.235	0.753	0.622

and December 2025 and spanning diverse technical domains under the IPC taxonomy. We additionally construct a scientific paper corpus from publicly accessible scholarly resources to evaluate the generalizability of FlowPIE across different literature sources. For each corpus, we extract abstracts, technical features, semantic embeddings, and citation metadata to construct the literature graph following Section 3.1. Further corpus statistics and construction details are provided in Appendix B.2.

Baselines. We compare FlowPIE with two types of baselines: ***(i)* LLM-based Framework**, SCIPIP (Wang et al., 2024), which leverages a dual-path framework that integrates retrieved literature with LLM-based brainstorming; and ***(ii)* Agent-based Framework**, including Research Agent (Baek et al., 2025), which iteratively uses a research agent and a review agent, and Chain-of-Ideas (Li et al., 2025), which employs a CoI-Agent to model dependency relations among prior works. We also compare against the multi-agent baseline VirSci (Su et al., 2025), which supports discovery through simulated team construction and structured discussion. To ensure a fair comparison, we use `GPT-4o-mini` as the idea generator for all methods. Implementation details and method costs are provided in Appendix B.1. We also provide details on the literature used for idea generation in Appendix B.2.

Evaluation Benchmarks and Metrics. We evaluate all methods on two SIG benchmarks. ***(i)* AI Idea Bench 2025** (Qiu et al., 2025) comprises papers from top AI conferences, including ICLR, CVPR, and ACL, as target idea sources. It contains three main tasks: idea-to-topic matching (I2T), idea-to-idea matching (I2I), and idea multiple-choice evaluation (IMCQ). The first two tasks use an LLM-as-a-judge paradigm with scores ranging from 1 to 5, while IMCQ is evaluated by accuracy. ***(ii)* IdeaBench** (Guo et al., 2025) contains 2,374 influential biomedical papers and evaluates generated ideas using two similarity-based metrics: BERTScore for semantic similarity, with a practical upper limit of 0.718 reported in the original paper, and idea overlap on a 0–10 scale. It also uses two insight scores for novelty and feasibility, computed from the relative ranking of generated ideas against the target paper's idea. For fair and consistent evaluation, all metrics are assessed using the frontier `GPT-5-mini` model. Detailed task and metric definitions are provided in Appendix D.1.

Human Evaluation Setup. We follow the criteria of Si et al. (2025), employing human experts who are computer science PhD students to blindly evaluate 20% randomly sampled ideas from AI Idea Bench 2025 for each method on novelty, feasibility, excitement, and expected effectiveness using a 10-point scale. Details are provided in Appendix D.3.

4.2 Results

Benchmark Results. As shown in Table 1 and Table 2, we report results on AI Idea Bench 2025 and IdeaBench, respectively. Across the three tasks of AI Idea Bench 2025, FlowPIE generates ideas with high consistency to the target topic and strong relevance to the target paper's idea. It is also the only method to obtain a motivation score above 4 in the I2I task. In the IMCQ task, FlowPIE achieves a motivation selection accuracy of 0.780 and an experimental plan selection accuracy of 0.635 when selecting the best idea among competing alternatives.

For IdeaBench, FlowPIE achieves the highest Semantic Similarity and Idea Overlap with the target paper. Considering the Novelty Insight Score (NI) and Feasibility Insight Score (FI), both FlowPIE

and its initial population lie on the Pareto front, indicating a competitive balance between novelty and feasibility. In particular, FlowPIE achieves a well-balanced trade-off between NI and FI.

Overall, FlowPIE achieves superior performance on both benchmarks and lies on the Pareto frontier. Notably, even the initial population of FlowPIE surpasses strong baselines such as SCIPIP. We report the standard deviation (*std*) for each task in Table 1 to support a more reliable evaluation. The consistently lower *std* of FlowPIE indicates that it generates ideas with greater robustness and more consistently high quality. We further provide additional results in Appendix E.1 using `Qwen2.5-7B` and `LLaMA3.1-8B` as backbones, demonstrating that our method generalizes across different LLMs. Although absolute performance is bounded by the capability of the underlying model, our method consistently yields stable relative improvements.

Table 2: **IdeaBench results.** *Simi.* denotes Semantic Similarity, *Overlap* denotes Idea Overlap. *NI* and *FI* denote Novelty and Feasibility Insight scores.

Method	Simi.	Overlap	NI	FI
SCIPIP	0.526	5.03	0.816	0.133
Research Agent	0.558	6.66	0.722	**0.138**
Chain-of-Ideas	0.482	7.24	**0.926**	0.095
VirSci	0.521	6.24	0.716	0.075
FlowPIE (*Ours*)	**0.559**	**7.76**	0.825	0.105
Initial Population	0.532	7.64	0.750	0.136

Reward Performance. As shown in Table 3, we evaluate all generated ideas on AI Idea Bench 2025 using the GRM, where reward computation is aligned with the fitness evaluation in our evolution process. We use `DeepSeek-V3.2` as the GRM backbone. The final idea population of FlowPIE achieves the highest average reward score among all baselines, with its initial idea population already outperforming competing methods. This initial population is generated using a threshold-based stopping criterion to ensure sufficient exploration, rather than a fixed budget or step size. Additionally, we visualize the reward lifecycle of FlowPIE in Figure 2, which shows a clear reward scaling trend. The rewards of the initial ideas increase with exploration steps, although they eventually reach a bottleneck despite remaining higher than those of other baselines. Subsequent idea evolution from this initial population substantially improves the rewards and shifts the distribution toward higher scores, as reflected by the continuously decreasing standard deviation. We conduct correlation analysis between reward and benchmark results in Appx. E.2.

Domain Generalization. Given the multi-domain nature of IdeaBench, which encompasses eight fields spanning Health & Medicine (HM), Genetics & Molecular Biology (GMB), Environmental Sciences (ES), Neuroscience & Cognitive Sciences (NCS), Technology & Engineering (TE), Social & Behavioral Sciences (SBS), Materials Science (MS), and Public Health & Policy (PHP), together with the AI domain covered by AI Idea Bench 2025, we report domain-specific rewards in Table 4. The results demonstrate the strong domain generalization of FlowPIE, which achieves the highest rewards across all nine domains and performs particularly well in Materials Science.

Table 3: Reward and human evaluation results. N: Novelty, F: Feasibility, E: Excitement, EE: Expected Effectiveness, and Avg.: Average.

Method	Reward			Human Evaluation				
	N	F	Avg.	N	F	E	EE	Avg.
SCIPIP	0.50	0.73	0.62	0.38	**0.37**	0.30	0.29	0.34
Research Agent	0.49	0.66	0.58	0.22	0.19	0.13	0.13	0.17
Chain-of-Ideas	0.59	0.66	0.63	0.34	0.29	0.31	0.29	0.31
VirSci	0.60	0.65	0.63	0.27	0.17	0.20	0.20	0.21
FlowPIE (*Ours*)	**0.75**	**0.76**	**0.76**	**0.45**	0.36	**0.38**	**0.37**	**0.39**
Initial Population	0.68	0.68	0.68	–	–	–	–	–
w/o isolation island	0.72	0.73	0.73	–	–	–	–	–

Table 4: Domain-specific reward performance across nine domains.

Method	AI	HM	GMB	ES	NCS	TE	SBS	MS	PHP
SCIPIP	0.62	0.60	0.64	0.63	0.63	0.62	0.61	0.67	0.60
Research Agent	0.58	0.63	0.66	0.65	0.65	0.63	0.60	0.69	0.58
Chain-of-Ideas	0.63	0.67	0.71	0.71	0.70	0.70	0.66	0.73	0.67
VirSci	0.60	0.65	0.69	0.66	0.67	0.67	0.61	0.70	0.60
FlowPIE(*Ours*)	**0.75**	**0.81**	**0.80**	**0.83**	**0.81**	**0.77**	**0.76**	**0.88**	**0.78**
Initial Population	0.68	0.72	0.67	0.78	0.70	0.68	0.67	0.83	0.69

Human Evaluation Results. We report human evaluation results in Table 3, showing that FlowPIE generates higher-quality ideas than the baselines. Additionally, we report the inter-expert Spearman correlations for each metric, together with the correlation between average reward and average human score in Table 5. Our reward demonstrates strong alignment with human judgments, achieving Spearman correlations of 0.60 and 0.87 with the average human ratings

Table 5: Inter-expert agreement of human evaluation and correlation between our reward and the average human score on novelty and feasibility.

Categories	Metric	N	F	E	EE
Human and Human	Spearman ρ	0.49	0.54	0.32	0.47
Human and Reward	Spearman ρ	0.60	0.87	–	–

on novelty and feasibility, respectively. Notably, these correlations are comparable to or even higher than the agreement between individual experts, suggesting that the reward reliably captures aggregated human preferences.

4.3 Analysis

Literature Exploration Analysis (RQ1). Most prior works recognize the importance of retrieved literature for ideation, typically using keyword–semantic hybrid retrieval or entity-centric citation graphs to provide relevant scientific knowledge. However, they usually retrieve a fixed number of papers in a one-shot, static manner. As shown in Figure 4 (a), FlowPIE couples retrieval with generation to enable dynamic exploration in both breadth and depth. Most ideas rely on around three publications, while some incorporate more than seven or even ten during initial ideation. This variation is determined by idea quality rather than a fixed hyperparameter.

Idea Diversity Analysis (RQ2). We assess the diversity of generated ideas following Si et al. (2025), encoding them with all-MiniLM-L6-v2. We consider two ideas similar if their cosine similarity exceeds the threshold of 0.65. The results are shown in Figure 4 (b). Si et al. (2025) investigate test-time idea diversity scaling with an increasing number of generated ideas and find that simply increasing quantity often leads to repeated or duplicate ideas. FlowPIE achieves the highest diversity score, and even the diversity of its initial population outperforms Chain-of-Ideas and VirSci. This result demonstrates that dynamic literature exploration and isolation-island-enhanced idea evolution can improve diversity beyond simple count scaling at test time.

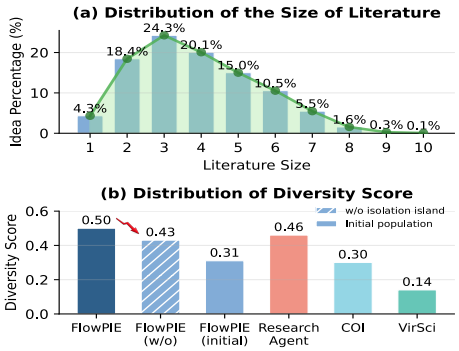

Figure 4: (a) Distribution of explored literature count; (b) Diversity score distribution.

4.4 Ablation Study

Ablations on flow-guided literature exploration. Prior LLM-enhanced EAs typically use zero-shot prompting or solutions from other methods as the initial population, but do not explicitly optimize the quality of this population. For example, EvoAgent (Yuan et al., 2025) initializes its agent population using MetaGPT (Hong et al., 2024). In contrast, our flow-guided MCTS dynamically explores the literature, filtering low-quality entries and weak ideas to construct a high-quality population for subsequent evolution. We conduct an ablation study by using ideas from SCIPIP and Chain-of-Ideas as the initial population, as reported in Table 6, which shows consistent performance improvements. This demonstrates the strong generalizability of our idea evolution process: it effectively enhances ideas generated by heterogeneous methods while achieving the largest gains under our own initialization strategy.

Ablations on idea evolution, *i.e. just initialization*. We conduct an ablation study by removing the idea evolution process and using only flow-guided MCTS to generate the initial population. As shown in Tables 1, 2, and 3, the initial idea population already outperforms other strong baselines, demonstrating the effectiveness of our initialization method. However, the initial ideas still exhibit a reward bottleneck despite increased exploration iterations. Figure 2 shows that increasing the number of exploration steps, corresponding to greater breadth and depth in the graph, yields only marginal gains, with the rewards of the initial ideas approaching a plateau.

Table 6: Idea evolution results with initial ideas from different baselines. We report the average reward (AR) and average score of three tasks in AI Idea Bench 2025.

Method	AR ↑	I2T ↑	I2I ↑	IMCQ ↑
SCIPIP	0.61	4.18	3.68	0.464
w/ Idea Evolution	0.63 (+3.3%)	4.24 (+1.4%)	3.70 (+0.5%)	0.475 (+2.4%)
Chain-of-Ideas	0.59	4.21	3.62	0.437
w/ Idea Evolution	0.64 (+8.5%)	4.23 (+0.5%)	3.63 (+0.3%)	0.448 (+2.5%)
FlowPIE (*initial*)	0.68	4.08	3.94	0.688
w/ Idea Evolution	0.75 (+10.1%)	4.18 (+2.5%)	4.15 (+5.3%)	0.708 (+2.9%)

Table 7: Demonstration of an idea generated by FlowPIE. Due to space limitations, we present only a concise summary of the final idea. The complete version of this idea is provided in Table 11.

Input topic: *The topic of this paper is improving reasoning capabilities in Large Language Models (LLMs).*
Final Idea Title: *Dynamic Macro-Guided Verification (DMGV): A Modular Reasoning Augmentation Framework for LLMs* **Summary**: Dynamic Macro-Guided Verification (DMGV) is a framework that converts recurrent reasoning subchains in CoT traces into **reusable reasoning macros** and attaches lightweight verifiers to validate their outputs. Instead of repeatedly regenerating similar reasoning steps, the model composes solutions by invoking validated macros and verifying their outputs. The core insight is that many reasoning traces contain repeated substructures that can be compressed into reusable operators. By validating macro outputs through interface-level verification rather than re-evaluating entire reasoning chains, DMGV aims to reduce both error accumulation and token-level redundancy in LLM reasoning. It consists of three stages: *(1) Macro discovery*: reasoning traces are segmented and clustered to identify recurrent subchains, which are distilled into parameterized macros with explicit input-output interfaces. *(2) Macro-guided reasoning*: during inference, the generator composes reasoning steps by invoking macros whenever a subgoal matches a macro signature. *(3) Targeted verification*: a lightweight verifier checks macro outputs using symbolic, arithmetic, or structural consistency tests; failed cases trigger localized fallback re-synthesis.

Ablations on isolation islands within mutation. We conduct an ablation study on the isolation islands within the mutation operator of idea evolution, i.e., we do not introduce OOD literature knowledge from the isolation islands of the current trajectory within graph. As shown in Figure 4 (b), idea diversity decreases compared with the full setting. The reward results under this ablation setting (Table 3, last row) only slightly underperform the full setting, suggesting that the isolation island strategy improves diversity while preserving overall quality.

Generalization across literature corpora. We evaluate FlowPIE across different literature corpora, including both patents and scientific papers. As shown in Table 1, FlowPIE consistently outperforms prior methods under both settings, while also improving over the corresponding initial populations across all metrics. Although the two corpora exhibit different strengths across individual metrics, both achieve strong overall performance. These results highlight the strong generalizability of FlowPIE across heterogeneous literature sources, beyond any specific literature corpus.

4.5 Qualitative Analysis

We present a case study on reasoning pattern abstraction for LLM reasoning in Table 7, showing how recurrent CoT subchains can be transformed into reusable and verifiable reasoning units, termed *reasoning macros*, which aim to reduce both error accumulation and token-level redundancy in LLM reasoning. Unlike prior modular reasoning approaches, this idea equips each macro with an explicit interface and lightweight verification, enabling reliable reuse, reducing repeated low-level generation, and localizing failures to individual macro calls. Additional cases across various domains are provided in Appendix G.

5 Conclusion

In this paper, we revisit the conventional scientific idea generation paradigm, in which literature is statically retrieved before idea generation, and propose **FlowPIE**, a novel framework for test-time idea evolution. FlowPIE integrates initial idea generation with dynamic literature exploration through flow-guided MCTS, while continuously refining ideas using evolutionary operators, including selection, crossover, and mutation. By using GRM-based rewards as feedback on initial idea quality, FlowPIE adaptively adjusts literature flow probabilities and guides exploration toward more informative trajectories. Benchmark results and human evaluation demonstrate that FlowPIE generates ideas that are more relevant, higher-quality, more stable, and more diverse than those produced by strong baselines, while also exhibiting robust generalization across both scientific domains and literature types. Beyond benchmark performance, reward analysis further shows clear test-time reward scaling, highlighting the effectiveness of coupling adaptive literature exploration with iterative idea evolution for autonomous scientific ideation.

AI USE STATEMENT

In this work, generative AI tools were used to assist with language polishing, translation, and code refinement and debugging. We determined and verified all research ideas, methodological choices, experimental settings, and conclusions, and manually reviewed all AI-assisted outputs for correctness. We take full responsibility for the final content.

REFERENCES

Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. Researchagent: Iterative research idea generation over scientific literature with large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6709–6738, 2025. URL https://aclanthology.org/2025.naacl-long.342/.

Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J Hu, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations. *Journal of Machine Learning Research*, 24(210):1–55, 2023. URL https://jmlr.org/papers/v24/22-0364.html.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the association for computational linguistics: ACL 2024*, pp. 2318–2335, 2024. URL https://aclanthology.org/2024.findings-acl.137.pdf.

Qiguang Chen, Mingda Yang, Libo Qin, Jinhao Liu, Zheng Yan, Jiannan Guan, Dengyun Peng, Yiyan Ji, Hanjing Li, Mengkang Hu, Yimeng Zhang, Yihao Liang, Yuhang Zhou, Jiaqi Wang, Zhi Chen, and Wanxiang Che. AI4Research: A Survey of Artificial Intelligence for Scientific Research. *arXiv e-prints*, art. arXiv:2507.01903, July 2025. doi: 10.48550/arXiv.2507.01903.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Petar Sirkovic, Artiom Myaskovsky, Grzegorz Glowaty, Felix Weissenberger, Alessio Orlandi, Dan Popovici, et al. Accelerating scientific discovery with co-scientist. *Nature*, pp. 1–3, 2026.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ZG3RaNIsO8.

Sikun Guo, Amir Hassan Shariatmadari, Guangzhi Xiong, Albert Huang, Myles Kim, Corey M Williams, Stefan Bekiranov, and Aidong Zhang. Ideabench: Benchmarking large language models for research idea generation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5888–5899, 2025. URL https://dl.acm.org/doi/10.1145/3711896.3737419.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VtmBAGCN7o.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. URL https://arxiv.org/abs/2410.21276.

Kuang-Huei Lee, Ian Fischer, Yueh-Hua Wu, Dave Marwood, Shumeet Baluja, Dale Schuurmans, and Xinyun Chen. Evolving Deeper LLM Thinking. *arXiv e-prints*, art. arXiv:2501.09891, January 2025. doi: 10.48550/arXiv.2501.09891.

Long Li, Weiwen Xu, Jiayan Guo, and et al. Chain of ideas: Revolutionizing research via novel idea development with LLM agents. In Christos Christodoulopoulos, Tanmoy Chakraborty,

Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 8971–9004, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.477. URL https://aclanthology.org/2025.findings-emnlp.477/.

Yusheng Liao, Shuyang Jiang, Zhe Chen, Yu Wang, and Yanfeng Wang. MedCare: Advancing medical LLMs through decoupling clinical alignment and knowledge aggregation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10562–10581, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.619. URL https://aclanthology.org/2024.findings-emnlp.619/.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. URL https://arxiv.org/abs/2412.19437.

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

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

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, Alhussein Fawzi, Josh Grochow, Andrea Lodi, Jean-Baptiste Mouret, Talia Ringer, and Tao Yu. Mathematical discoveries from program search with large language models. *Nature*, 625:468 – 475, 2023. URL https://api.semanticscholar.org/CorpusID:266223700.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. Can LLMs generate novel research ideas? a large-scale human study with 100+ NLP researchers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=M23dTGWCZy.

Haoyang Su, Renqi Chen, Shixiang Tang, Zhenfei Yin, Xinzhe Zheng, Jinzhe Li, Biqing Qi, Qi Wu, Hui Li, Wanli Ouyang, et al. Many heads are better than one: Improved scientific idea generation by a llm-based multi-agent system. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28201–28240, 2025. URL https://aclanthology.org/2025.acl-long.1368/.

Jiabin Tang, Lianghao Xia, Zhonghang Li, and Chao Huang. AI-researcher: Autonomous scientific innovation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=kQWyOYUAC4.

Wenxiao Wang, Lihui Gu, Liye Zhang, Yunxiang Luo, Yi Dai, Chen Shen, Liang Xie, Binbin Lin, Xiaofei He, and Jieping Ye. Scipip: An llm-based scientific paper idea proposer. *arXiv preprint arXiv:2410.23166*, 2024. URL https://arxiv.org/abs/2410.23166.

Wanghan Xu, Yuhao Zhou, Yifan Zhou, Qinglong Cao, Shuo Li, Jia Bu, Bo Liu, Yixin Chen, Xuming He, Xiangyu Zhao, et al. Probing scientific general intelligence of llms with scientist-aligned workflows. *arXiv preprint arXiv:2512.16969*, 2025. URL https://arxiv.org/abs/2512.16969.

Tian Ye, Zicheng Xu, Yuanzhi Li, and Zeyuan Allen-Zhu. Physics of language models: Part 2.1, grade-school math and the hidden reasoning process. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Tn5B6Udq3E.

Siyu Yuan, Kaitao Song, Jiangjie Chen, Xu Tan, Dongsheng Li, and Deqing Yang. EvoAgent: Towards automatic multi-agent generation via evolutionary algorithms. In Luis Chiruzzo, Alan

Ritter, and Lu Wang (eds.), *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)*, pp. 6192–6217, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.315. URL https://aclanthology.org/2025.naacl-long.315/.

A More Related Work Discussion

Scientific Idea Generation Evaluation and Benchmarks. As LLMs continue to demonstrate escalating performance across diverse tasks, there has been a notable proliferation of specialized frameworks designed for idea generation, To effectively measure the efficacy of these systems, several rigorous benchmarks, such as Scientist-Bench (Tang et al., 2025), AI Idea Bench 2025 (Qiu et al., 2025), IdeaBench (Guo et al., 2025) and SGI-Bench Xu et al. (2025). These benchmarks provide the necessary empirical groundwork to validate the substantive quality of proposed ideas, ensuring they are assessed not only for their surface-level novelty but also for their underlying feasibility and technological grounding.

In this paper, we adopt AI Idea Bench 2025 and Ideabench due to their suitability for our specific idea generation task and their inclusion of multi-domain attributes. These two benchmarks derive their ideas from recent high-quality publications, such as top conferences in the AI domain, providing golden reference ideas. In contrast, SGI-Bench is based on *Science's 125 Big Questions*, which do not have explicit optimal idea answers. Additionally, SGI-Bench provides the literature related to each idea in advance, which limits the evaluation of our method's literature exploration ability. For Scientist-Bench, idea generation is only a subset or starting point of its overall task, which focuses on a full-stage automated research pipeline, including experiment execution, code generation, and paper writing, differing from our research objective.

Due to the complexity and subjective nature of evaluating idea generation, these benchmarks mainly include two types of metrics: LLM-as-a-judge paradigm scores and similarity-based scores. For our experiments on AI Idea Bench 2025 and IdeaBench, which provide golden idea references, we can compute similarity-based metrics, such as I2T and I2I tasks in AI Idea Bench 2025, and Semantic Similarity and Idea Overlap in IdeaBench. They also provide LLM-as-a-judge metrics, such as the IMCQ task in AI Idea Bench 2025 and the Novelty Insight Score and Feasibility Insight Score in IdeaBench. These metrics do not rely on a single scalar score from an LLM; instead, they evaluate the relative quality of generated ideas from pairwise and listwise perspectives.

Although SGI-Bench proposes a combined objective and subjective metric for this task, its objective component heavily relies on the provided related literature. This makes it unfair for frameworks that can retrieve literature themselves, such as SCIPIP, Chain-of-Thought, and our FlowPIE. The objective metric is more suitable for evaluating the scientific information integration ability of the backbone model.

Additionally, the idea formats in prior works vary significantly. For example, Chain-of-Ideas (Li et al., 2025) structures ideas with a title, motivation, method, and experimental plan, whereas SCIPIP (Wang et al., 2024) includes only problem and method sections. Our work builds on the Chain-of-Ideas and Research Agent (Baek et al., 2025) formats, including motivation, method, and experimental plan. These different idea formats also cause the evaluation to be unstable. We believe that unifying the idea format and constructing standard evaluation metrics for these idea generation benchmarks is very important. Even though our evaluation setup includes GRM-based rewards, we consider this standardization an important direction for future work in the community.

Evolutionary Algorithm with LLMs. Evolutionary algorithms (EAs) are usually used for optimization problem search. With the advancement of LLMs, they can be leveraged as both an optimizer and a generator in EAs, thanks to their extensive out-of-domain knowledge and ability to operate in semantically rich language spaces. Researchers have already explored incorporating LLMs into EAs frameworks. Guo et al. (2024) propose EvoPrompt, which combines an LLM-enhanced evolutionary algorithm framework with discrete prompt optimization. Yuan et al. (2025) propose the EvoAgent framework, which treats existing agent frameworks as initial individuals and applies a series of evolutionary operators, significantly improving performance on writing, code, and travel planning tasks. Notably, Lee et al. (2025) introduce a framework called Mind Evolution, which

Algorithm 1 Initial Idea Population Construction via Flow-Guided MCTS

Input: Query q, literature graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, maximum iteration N, convergence threshold ϵ
Output: Initial idea set $\mathcal{I}$

Initialize root node $s_0 \leftarrow q$, global flow $F(s_0) \leftarrow 1$
    // Treat the query as the root of literature exploration.
Retrieve top-k relevant literature as children of s_0
    // Initialize the starting literature nodes via semantic retrieval.
Initialize $\mathcal{I} \leftarrow \varnothing$, reward buffer $\mathcal{R} \leftarrow \varnothing$, $n \leftarrow 0$
while $n < N$ **and** $\mathrm{Var}(\mathcal{R}) > \epsilon$ **do**
    $n \leftarrow n + 1$, $s \leftarrow s_0$, $\phi \leftarrow [s_0]$ // Start a new literature exploration trajectory.
    while $A(s) \neq \varnothing$ **do**
        $s' \leftarrow \arg\max_{u \in A(s)} UCB(u \mid s)$ // Select next node via flow-guided UCB (Eq. 1).
        Append s' to ϕ, set $s \leftarrow s'$
        if s is expandable and not fully explored **then**
            Expand one adjacent literature node into $A(s)$ // Expand a new literature node for exploration.
            break
        end if
    end while
    $i \leftarrow \mathrm{GenerateIdea}(\phi)$ // Generate a scientific idea conditioned on the literature trajectory.
    $R \leftarrow \mathrm{GRM}(i)$ // Evaluate the idea using the generative reward model.
    $\mathcal{I} \leftarrow \mathcal{I} \cup \{i\}$, $\mathcal{R} \leftarrow \mathcal{R} \cup \{R\}$
    for each transition $(s_t \rightarrow s_{t+1})$ in ϕ **do**
        Compute $\tilde{R}_t = R \cdot \gamma^{T-t}$ // Apply depth-aware reward decay.
        Update $Q(s_{t+1} \mid s_t)$ by running average
        Update $P_f(s_{t+1} \mid s_t)$ using Eq. 2
        Normalize $P_f(\cdot \mid s_t)$ over $A(s_t)$
        Update global flow $F(s_{t+1}) = F(s_t) \cdot P_f(s_{t+1} \mid s_t)$
        // Maintain consistency with global flow constraint.
    end for
end while
return $\mathcal{I}$ // Generated ideas form the initial population for the evolution stage.

leverages evolutionary algorithms to scale LLMs' test-time computation, outperforming Best-of-N and Sequential Revision test-time reasoning strategies at the same computational cost.

Our FlowPIE builds on EAs for test-time computation, leveraging multiple operators to refine the idea population based on the LLM-based idea generator. Additionally, prior works do not focus on evolving a high-quality initial population. Most use zero-shot results as the starting population or rely on results from other methods, such as EvoAgent. We argue that successful evolution depends on a strong initial parent idea population. Therefore, we propose a novel method for constructing the initial idea population, combining dynamic literature exploration and initial idea generation with GFlowNet-inspired flow-guided MCTS, balancing exploration and exploitation of the literature.

B Experimental Setting

B.1 Implementation Details

We implement our FlowPIE using publicly available LLMs for both idea generation and evaluation. Specially, we use `GPT-4o-mini` as the backbone model for idea generator, accessed through the official API. For fitness evaluation and idea quality analysis, we employ `DeepSeek-V3.2` as the fitness evaluator. For flow-guided MCTS, the exploration coefficient c in Eq. 1 is set to commonly used value $c = \sqrt{2}$ for balancing exploration and exploitation. The learning rate α for updating the flow probability P_{flow} in Eq. 2 is set to 0.2. For the test-time Idea Evolution, the maximum number of evolution iterations is limited to 20. Additionally, if the standard deviation of the generated ideas' fitness score is lower than threshold 0.05, the evolution process terminates as the final ideas.

Method Cost. We report the API cost of each method for generating a scientific idea, where `GPT-4o-mini` is used as the LLM-based idea generator across all methods, as shown in Table 8.

Table 8: API cost comparison for idea generation across different methods. All methods use GPT-4o-mini as the idea generator.

Method	Prompt Tokens	Avg. Tokens / Idea	Avg. Cost / Idea
SCIPIP	4062.50	921.33	$0.038
Research Agent	1755.44	477.53	$0.028
Chain-of-Ideas	4526.92	650.43	$0.052
Virsci	1458.36	789.60	$0.061
FlowPIE (*Ours*)	4123.43	924.25	$0.056

B.2 Literature Database Details

Literature Source and Data Statistics. Our literature dataset consists of both patent documents and scientific publications collected from publicly available sources. The patent subset is obtained from the USPTO[1], containing 16,050 patents published between October and December 2025, while the paper subset is collected from publicly accessible scholarly resources. The resulting corpus covers diverse scientific and technical domains. Table 9 reports the distribution of the patent subset across major International Patent Classification (IPC) sections, illustrating its technological coverage.

Table 9: Distribution of IPC sections. IPC sections: A for Human Necessities; B for Performing Operations and Transporting; C for Chemistry and Metallurgy; D for Textiles and Paper; E for Fixed Constructions; F for Mechanical Engineering, Lighting, Heating, Weapons, and Blasting; G for Physics; H for Electricity.

Section	Count	Percentage (%)
A	1293	8.06
B	980	6.11
C	710	4.42
D	26	0.16
E	123	0.77
F	208	1.30
G	7957	49.58
H	4753	29.61
All	**16050**	**100**

Data Collection and Processing. The raw literature is first parsed into structured textual components, including abstracts, technical content, and citation metadata. To facilitate structured reasoning over the literature corpus, we employ a LLM to extract key semantic attributes from each document.

Following the formulation introduced in Section 3.1, each document $p \in \mathcal{D}$ is mapped into an attribution tuple $\langle \mathcal{A}_p, \mathcal{F}_p, \mathcal{S}_p \rangle$. Where $\mathcal{A}_p$ represents its abstract, is directly extracted from the structured document metadata. The $\mathcal{F}_p$ represents the core technical feature, which is generated by a LLM through analyzing the full document. The extracted features are further verified by human experts to ensure their correctness and completeness. The $\mathcal{S}_p$ denotes its semantic embedding, which is obtained by encoding the document using `BGE-M3` (Chen et al., 2024) as the embedding model.

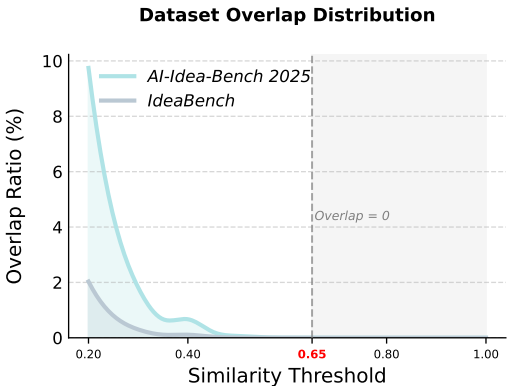

Figure 5: Dataset overlap distribution between our literature and two benchmarks across different similarity levels. The overlap ratio rapidly decreases as similarity increases and approaches zero in the high-similarity region, indicating minimal risk of data leakage.

[1] https://data.uspto.gov/home

Algorithm 2 Test-Time Idea Evolution

Input: Initial idea population $\mathcal{I}_0$ of size N, literature graph $\mathcal{G}$, offspring number M, mutation rate ρ, maximum evolution step T, convergence threshold ϵ
Output: Final evolved idea population $\mathcal{I}^*$

Initialize population $\mathcal{I} \leftarrow \mathcal{I}_0$, evolution step $t \leftarrow 0$
    // Use the ideas from Algorithm 1 as the starting population.
Evaluate all ideas in $\mathcal{I}$ with GRM to obtain fitness scores
    // Fitness computation on novelty and feasibility.
while $t < T$ **and** $\Delta\text{Reward}(\mathcal{I}) > \epsilon$ **do**
    $t \leftarrow t + 1$, $\mathcal{O} \leftarrow \varnothing$ // Generate a new offspring set at each evolution step.
    for $m = 1$ to M **do**
        Select two parent ideas i_a, i_b from $\mathcal{I}$
        $o \leftarrow \text{Crossover}(i_a, i_b)$ // Recombine core technical features of two promising parent ideas.
        if $\text{Bernoulli}(\rho)$ **then**
            $\mathcal{L}_{ood} \leftarrow \text{IsolationIsland}(\mathcal{G}, i_a, i_b)$
            $o \leftarrow \text{Mutate}(o, \mathcal{L}_{ood})$
        // Inject literature from isolated island to encourage conceptual diversity.
        end if
        Evaluate offspring o with GRM to obtain fitness $R(o)$
        $\mathcal{O} \leftarrow \mathcal{O} \cup \{o\}$
    end for
    $\mathcal{C} \leftarrow \mathcal{I} \cup \mathcal{O}$ // Merge parent and offspring ideas into a candidate pool.
    $\mathcal{I}_{next} \leftarrow \varnothing$
    while $|\mathcal{I}_{next}| < N$ **do**
        Sample a random subset $\mathcal{S} \subset \mathcal{C}$
        $i^* \leftarrow \arg\max_{i \in \mathcal{S}} R(i)$
        $\mathcal{I}_{next} \leftarrow \mathcal{I}_{next} \cup \{i^*\}$
        $\mathcal{C} \leftarrow \mathcal{C} \setminus \{i^*\}$ // Tournament selection preserves high-fitness ideas for the next generation.
    end while
    $\mathcal{I} \leftarrow \mathcal{I}_{next}$
end while
return $\mathcal{I}$ // Return the final evolved population after convergence.

Data Quality Analysis. We further perform data quality analyses to confirm that our literature database does not suffer from knowledge leakage. Following Si et al. (2025), we encode all literature (papers and patents) using `all-MiniLM-L6-v2` and compute the cosine similarity between our literature corpus and the target papers in the two benchmarks. We provide the similarity distribution as shown in Figure 5. Most similarity scores fall below the threshold 0.3, with none literature approaching the predefined duplication threshold (0.65). This indicates limited semantic overlap between the two corpora, suggesting that our literature database is largely independent of the benchmark datasets and poses minimal risk of data leakage.

C Methodology Details

In this section, we provide further details on the motivation and formulation of the proposed flow-guided MCTS, including the flow-guided UCB for forward selection and the backpropagation mechanism. The detailed algorithms are summarized in Algorithm 1 and Algorithm 2.

C.1 Flow-Guided Exploration Formulation

We formulate literature exploration as a sequential retrieval process represented by a search tree. Each node corresponds to a partial state s, and each edge represents an expansion action producing a successor state $s' \in A(s)$.

In standard MCTS, action selection relies on empirical value estimation:

$$Q(s'|s) = \frac{1}{N(s'|s)} \sum_{i=1}^{N(s'|s)} R_i \tag{3}$$

where $N(s)$ and $N(s'|s)$ denote node and edge visitation counts, respectively. Classical UCB-based selection encourages exploration via uncertainty-driven bonuses:

$$UCB(s'|s) = Q(s'|s) + c\sqrt{\frac{lnN(s)}{1+N(s'|s)}} \tag{4}$$

while effective for optimization problems, this mechanism typically concentrates probability mass onto a single dominant trajectory. However, scientific idea generation requires discovering multiple high-quality solutions rather than a single optimum.

To address this limitation, our FlowPIE introduces a global flow variable that redistributes exploration probability across the search tree. FlowPIE associate each state with a non-negative flow quantity $F(s)$, representing the amount of exploration mass reaching the state. The root node is initialized as $F(s_0) = 1$ The outgoing flow is decomposed over candidate expansions:

$$F(s'|s) = F(s)P_f(s'|s) \tag{5}$$

where $P_f(s'|s)$ denotes the relative importance of transition $s \to s'$. This induces the conservation constraint:

$$\sum_{s_i \in A(s)} P_f(s_i|s) = 1 \tag{6}$$

Therefore, we define a probability distribution governing exploration allocation. Replacing uniform exploration in classical UCB with flow-aware allocation yields the proposed **Flow-Guided UCB**:

$$UCB(s'|s) = Q(s'|s) + c \cdot P_f(s'|s)\frac{\sqrt{N(s)}}{1+N(s'|s)} \tag{7}$$

Intuitively, transitions receiving larger global flow probability obtain stronger exploration preference, allowing promising reasoning directions to be revisited without collapsing exploration diversity. The flow probability propagates recursively along trajectories $F(s_{i+1}) = F(s_i)P_f(s_{i+1}|s_i)$, enabling downstream states to inherit global importance.

C.2 Reward-Driven Flow Update

Given a rollout terminating with reward R, temporal credit assignment is performed using a decay mechanism $\tilde{R}_t = R \cdot \gamma^{T-t}$, where T denotes the maximum depth of trajectory, γ is a discount factor. Transition flow is updated using exponential averaging:

$$P_f(s'|s) \leftarrow (1-\alpha)P_f(s'|s) + \alpha\tilde{R}_t \tag{8}$$

Under this update rule, transitions belonging to high-reward trajectories continuously accumulate larger incoming flow mass.

Consequently, the proposed search process approximates the core principle of Generative Flow Networks, encouraging sampling from a distribution of high-reward solutions rather than converging to a single optimal path.

D Evaluation Details

D.1 Benchmark Tasks and Metrics

As mentioned in Section 4.1, we adopt two benchmarks including AI Idea Bench 2025 and IdeaBench. In this section, we provide detailed descriptions of their tasks and metrics to facilitate a better understanding of the experimental results.

AI Idea Bench 2025. This benchmark comprises three main tasks: idea-to-topic matching (I2T), idea-to-idea matching (I2I), and idea multiple-choice evaluation (IMCQ). The first two tasks mainly assess similarity and relevance, measuring the alignment between an idea and the topic, and between an idea and a reference idea. Both tasks leverage an LLM-simulated similarity function to assign scores on a 0–5 scale.

Human Evaluation Criteria: Novelty

Definition and Guideline: Whether the idea is creative and different from existing works on the topic, and brings fresh insights. You are encouraged to search for related works online. You should consider all papers that appeared online prior to July 2024 as existing work when judging the novelty.

Scores
[1] Not novel at all: there are many existing ideas that are the same
[2]
[3] Mostly not novel: you can find very similar ideas
[4]
[5] Somewhat novel: there are differences from existing ideas but not enough to turn into a new paper
[6] Reasonably novel: there are some notable differences from existing ideas and probably enough to turn into a new paper
[7]
[8] Clearly novel: major differences from all existing ideas
[9]
[10] Very novel: very different from all existing ideas in a very interesting and clever way

Human Evaluation Criteria: Feasibility

Definition and Guideline: How feasible it is to implement and execute this idea as a research project? Specifically, how feasible the idea is for a typical CS PhD student to execute within 1-2 months of time. You can assume that we have abundant OpenAI / Anthropic API access, but limited GPU compute.

Scores
[1] Impossible: the idea doesn't make sense or the proposed experiments are flawed and cannot be implemented
[2]
[3] Very challenging: there are flaws in the proposed method or experiments, or the experiments require compute/human resources beyond any academic lab
[4]
[5] Moderately feasible: It can probably be executed within the given time frame but would quire careful planning, efficient use of APIs or some advanced computational strategies to overcome the limited GPU resources, and would require some modifications to the original proposal to make it work
[6] Feasible: Can be executed within the given constraints with some reasonable planning
[7]
[8] Highly Feasible: Straightforward to implement the idea and run all the experiments
[9]
[10] Easy: The whole proposed project can be quickly executed within a few days without requiring advanced technical skills

Figure 6: Human evaluation criteria. Top: The definition and guideline of novelty score. Bottom: The definition and guideline of feasibility score.

- **I2T**: Evaluate whether the generated idea aligns with the specified topic and assess its degree of alignment with the target paper's topic, with a LLM-based similarity function $F_{IT}^{D} : (I_B, T_{\text{topic}}) \rightarrow [0, 5]$, where measures the similarity between an idea $i \in I_B$ and the topic T_{topic} of the target paper. Finally, we leverage the equation below to compute the

alignment score of the current baseline B with respect to the topic:

$$S_{IT} = \max_{i_j \in I_B} F_{IT}^{D}(i_j, T_{\text{topic}}) \tag{9}$$

- **I2I**: Compare the generated idea with the motivation and experimental framework of the target paper using an LLM-based similarity function $F_{2I}^{D} : (I_B, (M_T, E_T)) \to [0, 5]$, where measures the similarity between an idea $i \in I_B$ and the motivation M_T and experimental framework E_T. Finally, we compute the score of the current baseline B using the equation below:

$$S_{I^2} = \max_{i_j \in I_B} F_{2I}^{D}(i_j, (M_T, E_T)) \tag{10}$$

The final task IMCQ focuses on evaluating the quality of generated ideas by constructing a multiple-choice question for each query. Each question comprises four options, where L_1 and L_2 are drawn from influential prior work exhibiting the closest conceptual alignment with the target paper T and the L_3 is the paper that maintains the highest degree of similarity to all target papers in the dataset. Finally, the baseline-generated idea is incorporated as the answer option L_4 to form the complete option set $C = \{L_1, L_2, L_3, L_4\}$. The answers of each question $R_B = \{r_1, r_2, \ldots, r_n\}$ produced by baseline B are compared with the correct option set A_c (corresponding to baseline's idea) to compute an accuracy-based score:

$$S_M = \begin{cases} 1, & \text{if } A_c \cap R_B \neq \varnothing, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

The selected option is regarded as the best among the four candidates. A higher accuracy indicates that our idea is selected more frequently, reflecting stronger capability of the baseline.

IdeaBench. This benchmark provides two types of metrics: similarity and Insight Score, with the latter assessing novelty and feasibility. The similarity-based metrics comprise two components: semantic similarity, evaluated using the F1 score of BERTScore with DeBERTa-xlarge-MNLI, and idea overlap, which measures the overlap between the generated idea and the abstract of the target paper on a 0–10 scale, accompanied by an explanation. Notably, the original paper reports a practical upper limit of 0.718 for semantic similarity.

The Insight Score is evaluated from a ranking perspective, which combined m ideas from target papers and n generated ideas from baselines. For each query, we leverage an LLM to rank the n generated ideas together with one target idea. If all generated ideas are ranked higher than the target idea, we assign $r_{\text{target}_i|q} = n + 1$. Conversely, if the target idea outperforms all generated ideas, its rank is $r_{\text{target}_i|q} = 1$. Otherwise, we use its actual ranking position to compute the Insight Score:

$$I(LLM, q) = \frac{1}{m} \sum_{i=1}^{m} \frac{r_{\text{target}_i|q} - 1}{n} \tag{12}$$

Where the novelty and feasibility Insight Scores depend on the specific evaluation prompt. During benchmark evaluation, to ensure stable scoring results and reduce randomness in model outputs, we set the LLM judge's decoding temperature to 0.2.

D.2 Reward Metrics

In our FlowPIE, we primarily use novelty and feasibility as reward metrics in the generative reward model, which assigns 1–5 scale scores with a reasoning chain-of-thought to provide interpretable justifications. Novelty measures the degree to which a generated idea introduces new scientific concepts or combinations beyond the existing literature, reflecting its potential to contribute original innovations. Feasibility evaluates whether the proposed idea is technically implementable and logically consistent given current technological capabilities. These two metrics are widely adopted in prior research on automated scientific idea generation such as SCIPIP (Wang et al., 2024) and Research Agent (Baek et al., 2025) also use novelty and feasibility as key criteria to assess the quality of ideas, balancing originality with practical viability. The detailed reward criterias and LLM evaluation prompts are provided in Figure 11.

Human Evaluation Criteria: Excitement

Definition and Guideline: How likely the proposed idea is going to work well (e.g., better than existing baselines).

Scores

[1] Poor: You cannot identify the contributions of this idea, or it's not interesting at all and you would fight to have it rejected at any major AI conference
[2]
[3] Mediocre: this idea makes marginal contributions and is very incremental
[4]
[5] Leaning negative: it has interesting bits but overall not exciting enough
[6] Learning positive: exciting enough to be accepted at a major AI conference, but still has some weaknesses or somewhat incremental
[8] Exciting: would deepen the community's understanding or make major progress in this research direction
[9]
[10] Transformative: would change the research field profoundly and worth a best paper award at major AI conferences

Human Evaluation Criteria: Expected Effectiveness

Definition and Guideline: How feasible it is to implement and execute this idea as a research project? Specifically, how feasible the idea is for a typical CS PhD student to execute within 1-2 months of time. You can assume that we have abundant OpenAI / Anthropic API access, but limited GPU compute.

Scores

[1] Extremely Unlikely: The idea has major flaws and definitely won't work well
[2]
[3] Low Effectiveness: The idea might work in some special scenarios but you don't expect it to work in general
[4]
[5] Somewhat ineffective: There might be some chance that the proposed idea can work better than existing baselines but the improvement will be marginal or inconsistent
[6] Somewhat effective: There is a decent chance that the proposed idea can beat existing baselines by moderate margins on a few benchmarks
[7]
[8] Probably Effective: The idea should offer some significant improvement over current methods on the relevant benchmarks
[9]
[10] Definitely Effective: You are very confident that the proposed idea will outperform existing methods by significant margins on many benchmarks

Figure 7: Human evaluation criteria. Top: The definition and guideline of excitement score. Bottom: The definition and guideline of expected effectiveness score.

D.3 Human Evaluation

Evaluation Criteria. We follow the human evaluation protocol for LLM-generated ideas proposed by Si et al. (2025), adopting a four-dimensional evaluation framework that includes novelty, feasibility, excitement, and expected effectiveness. We provide detailed evaluation guideline and criteria for human experts, including metric definitions and the interpretation of each score on the 1–10 scale, as shown in Figure 6 and 7.

Table 10: Performance comparison across different LLM backbones on the AI Idea Bench 2025.

Method	I2T	I2I	IMCQ
Backbone: GPT-4o-mini (released July 2024)			
SCIPIP	4.18±0.662	3.68±0.383	0.464
Research Agent	4.08±0.675	3.62±0.555	0.504
FlowPIE (*Ours*)	**4.18±0.544**	**4.15±0.333**	**0.708**
Backbone: Qwen2.5-7B-Instruct (released Sep. 2024)			
SCIPIP	4.02±0.781	3.80±0.434	0.463
Research Agent	3.98±0.539	3.76±0.412	0.442
FlowPIE (*Ours*)	**4.05±0.530**	**3.87±0.420**	**0.508**
Backbone: LLaMA3.1-8B-Instruct (released July 2024)			
SCIPIP	3.37±0.613	3.82±0.393	0.452
Research Agent	3.26±0.725	3.85±0.314	0.433
FlowPIE (*Ours*)	**3.68±0.554**	**3.89±0.448**	**0.486**

Details of Human Experts. We recruit three PhD students majoring in Computer Science, each of whom has published papers in top-tier AI conferences and has served as a reviewer, ensuring the expertise and reliability of the evaluation. We provide each evaluator with anonymized ideas in random order to ensure that they are unaware of their sources, thereby enabling a fair comparison.

E More Analysis

E.1 Effects of Different LLM Backbones

As shown in Table 10, we conduct additional experiments on AI Idea Bench 2025 using different backbone models, including Qwen2.5-7B-Instruct (released in September 2024) and LLaMA3.1-8B-Instruct (released in July 2024). We intentionally avoid using the most recent open-source models (e.g., the Qwen3.5 series) because of potential data leakage in scientific idea generation benchmarks, where model pretraining data may include papers from the test set. The consistently superior results across different backbone models demonstrate that FlowPIE is compatible with diverse LLMs. Although its performance is still bounded by the capability of the underlying model, FlowPIE consistently delivers improvements over the corresponding backbone.

E.2 Reward-Benchmark Consistency Analysis

To evaluate cross-benchmark consistency, we compute Kendall's W across all metrics and assess rank correlation between average reward and AI Idea Bench 2025. We observe moderate but statistically significant inter-metric agreement (Kendall's $W = 0.326$, p-value = 0.023), indicating partial yet non-random ranking consistency across complementary evaluation dimensions. In addition, average reward exhibits strong correlation with AI Idea Bench 2025 (Kendall's tau $\tau = 0.8667$, p-value = 0.0167), suggesting robust external alignment. Notably, FlowPIE consistently lies on the Pareto frontier, reflecting balanced multi-objective performance rather than metric-specific optimization.

F Prompts

Initial Idea Population Construction. We provide the initial idea generation prompt, as shown in Figure 8.

Parent Ideas Crossover. We provide the prompt for applying the crossover operator to parent ideas, as shown in Figure 9.

Idea Mutation. We provide the prompt for applying the mutation operator to generated ideas, as shown in Figure 10.

Initial Idea Generation Prompt

You are an expert in cross-domain innovation and idea generation. Your task is to generate a scientific idea that aligns with the user's scientific intent. The objective is to produce ideas that are technically specific, experimentally verifiable, and potentially claim-supportive rather than abstract or purely descriptive. The proposed idea should address a clear scientific problem, produce a **measurable and verifiable technical effect**, and demonstrate **non-trivial scientific value** that could be considered meaningful in a paper peer review context.

User Query:

{query}

And the following materials serve as background context and scientific inspiration. They should not be summarized, paraphrased, or directly reused.

Extracted technical elements

{technical_elements}

Reference literature metadata:

{ref_info}

Output Requirements: Generate one scientific idea together with a experimental validation plan. The output must contain the following five sections:

- **(A) Core Method Description:** Briefly summarize the proposed idea and the scientific problem it addresses.
- **(B) Functional Principle:** Describe the key scientific and functional principle. When appropriate, include 1{2 formulas or models representing the underlying logic.
- **(C) Concrete Workflow:** Provide a concrete workflow describing the operation of the proposed idea.
- **(D) Potential Innovation Directions:** List 3{5 potential scientific direction for future work.
- **(E) Experimental Design:** Design a experiment plan to validate the effect of the idea compared with baselines. The design could specify: Experimental Setup (such as Backbone Models, hyperparameter and so on), Variables (Independent variables, Dependent variables), Evaluation Metrics and Baselines.

The idea should focus on scientific principles and effects, rather than just step-by-step engineering implementation

Output Format

Return only the five sections (A{E) described above. The experimental design should clearly demonstrate how the proposed idea leads to measurable scientific improvements.

Figure 8: Prompt for the initial idea population generation.

Initial Idea Assessment and Fitness Evaluation using GRM. We provide the prompt used for novelty and feasibility assessment by the generative reward model, as shown in Figure 11. The fitness evaluation adopts the same prompt as that used in the GRM-based initial idea assessment, as shown in Figure 11.

G EXAMPLES

We provide the full version of our demonstration of the idea in Table 11, DMGV, generated by FlowPIE, whose topic is improving reasoning capabilities in large language models. The central idea is to abstract recurrent sub-reasoning patterns into reusable reasoning macros and improve the reliability of their reuse through lightweight verification. This directly targets a central challenge in multi-step LLM reasoning: *error propagation and accumulation in intermediate reasoning steps*, while repeatedly generating similar reasoning fragments is computationally inefficient. By combin-

Ideas Crossover Prompt

```
You are an expert innovation designer.  Your task is to intelligently
fuse two scientific ideas into one superior idea, leveraging the
provided literature context including both relevant literature and
diverse island literature.
Research Topic:
{topic}
Isolation Island Literature Context:
{literature_context}
Scientific Idea A(score:{idea_a.score}):
{idea_a.text}
Scientific Idea B(score:{idea_b.score}):
{idea_b.text}
Requirement:
1. Carefully analyze both ideas and identify their strengths
2. Study the literature context to understand existing science and
   identify gaps
3. Intelligently merge the best elements from both ideas
4. Leverage insights from island literature to enhance novelty and
   diversity
5. Create a NEW, COHERENT idea that combines their advantages
6. Ensure all five parts (A-E) are present and well-integrated
7. The new idea should be more innovative, feasible, and diverse than
   either parent
Output Format:
• (A) Core Method Description(Your fused core method)
• (B) Functional Principle(Your fused functional principles)
• (C) Concrete Workflow(Your fused workflow)
• (D) Potential Innovation Directions(Your fused innovation points)
• (E) Experimental Design(Your fused experimental design)
```

Figure 9: Prompt for the parent ideas crossover operator.

ing macro reuse with verifier-guided checking, the generated proposal offers a plausible direction for improving both reasoning accuracy and efficiency.

Additionally, we provide several concise summaries of generated ideas across multiple domains, particularly in Health & Medicine, Genetics & Molecular Biology, and Environmental Sciences, as shown in Table 12, 13 and 14.

Idea Mutation Prompt

```
You are an expert innovation improver.  Your task is to enhance
and mutate an existing scientific idea to make it more novel and
technically sound, leveraging diverse literature information.
Research Topic:
{topic}
Isolation Island Literature Context:
{literature_context}
Original Idea(score:idea.score):
{idea.text}
Requirement:
1. Identify potential weaknesses or areas for improvement
2. Study the literature context to find inspiration from both relevant
   and island literature
3. Introduce novel scientific elements or approaches inspired by
   diverse literature sources
4. Enhance the innovation level while maintaining feasibility
5. Improve clarity and scientific depth
6. Make sure all five parts (A-E) are strengthened
7. Incorporate insights from island literature to increase diversity
   and novelty
Output Format:
• (A) Core Method Description(Your improved core method)
• (B) Functional Principle(Your improved functional principles)
• (C) Concrete Workflow(Your improved workflow)
• (D) Potential Innovation Directions(Your improved innovation points)
• (E) Experimental Design(Your improved experimental design)
```

Figure 10: Prompt for the parent ideas mutation operator.

GRM Novelty Score Assessment Prompt

```
You are a literature examination expert, please evaluate the "novelty
of the scientific idea" for the following ideas.
You MUST output your result in valid JSON format only.
Idea content:
{idea.text}
Rating criteria:
Novelty(1-5) definition: How original and innovative the idea is
compared to existing research.
• 5: Extremely novel and groundbreaking. The idea introduces new,
  unexplored concepts or radically shifts the direction of the field.
• 4: Highly original. The idea is new and innovative but may still be
  building upon existing concepts or research.
• 3: Moderately original. The idea brings some new insights but is
  similar to existing work or follows well-established concepts.
• 2: Slightly original. The idea offers minor variations or
  incremental improvements to existing research but lacks substantial
  novelty.
• 1: Not original. The idea closely resembles existing research with
  little to no innovation.
Output Format:
{{ "novelty_score": <decimal number between 1 and 5> }}
```

GRM Feasibility Score Assessment Prompt

```
You are a literature examination expert, please evaluate the
"feasibility of the scientific idea" in terms of logic, consistency,
and combinatorial reasonableness for the following ideas.
You MUST output your result in valid JSON format only.
Idea content:
{idea.text}
Rating criteria:
Feasibility(1-5) definition: How realistic and practical the idea is
to implement in current scientific and technological conditions.
• 5: Fully feasible. The idea can be realistically executed
  with existing methods, data, and resources, and the plan for
  implementation is clear and practical.
• 4: Highly feasible. The idea is feasible with current technologies
  but may require some advancements or additional resources.
• 3: Moderately feasible. The idea faces significant practical
  challenges, requiring considerable advancements in technology or
  data.
• 2: Slightly feasible. The idea is difficult to implement with
  current resources and would need significant breakthroughs.
• 1: Not feasible. The idea is impractical and unlikely to be
  implemented with current technologies or methods.
Output Format:
{{ "feasibility_score": <decimal number between 1 and 5> }}
```

Figure 11: Prompt for the generative reward model for novelty and feasibility evaluation.

Table 11: Full demonstration of an idea generated by FlowPIE.

Input topic: *The topic of this paper is improving reasoning capabilities in LLMs.*

Final Idea: *Dynamic Macro-Guided Verification (DMGV): A Modular Reasoning Augmentation Framework for LLMs*

(A) Core Concept Description

Introduce a modular reasoning augmentation for LLMs called Dynamic Macro-Guided Verification (DMGV). DMGV automatically discovers, compresses, and reuses recurrent sub-reasoning patterns ("reasoning macros") from LLM chain-of-thought traces, and pairs those macros with a lightweight verifier that performs targeted, symbolic or model-based checks on macro outputs. At inference time the LLM's generator composes higher-level proofs by invoking validated macros (rather than re-generating low-level steps), and the verifier selectively validates macro applications and issues localized repairs. The invention addresses the technical problem of error accumulation and inefficiency in multi-step LLM reasoning: by (1) turning repeated sub-proofs into compact, validated operators, and (2) constraining verification to macro interfaces, DMGV reduces cumulative hallucination, lowers tokens/computation for repeated reasoning, and yields measurable improvements in final-answer accuracy and calibration.

(B) Conceptual Functional Principle

Key principles:

- **Pattern discovery:** across many proof traces, recurrent sub-chains $f_i(x) \approx y$ (where x are sub-problem inputs and y their subproof outcomes) are clustered and compressed into parameterized macros $m_j(\cdot;\theta_j)$ that approximate these sub-chains with bounded approximation error ε_j.
- **Interface verification:** each macro exposes a small, checkable interface I_j (inputs, claimed invariants, outputs). A verifier V executes a fast check $C_j : I_j \to \{\text{pass}, \text{fail}, \text{repair-hint}\}$ that tests logical consistency or symbolic constraints cheaply.
- **Composition with guarded application:** the generator G composes macros and raw steps. DMGV enforces that application of macro m_j is only accepted if either C_j passes or a bounded re-synthesis fallback is invoked.

A compact formalization:

Let G produce a chain-of-thought trace $T = [s_1, \ldots, s_n]$. Identify subchains $S_k = s_a \ldots s_b$ that map to function $f_k : X_k \to Y_k$. Learn macro m_j parameterizing a mapping $m_j(x;\theta_j) \approx f_k(x)$, with empirical error $\varepsilon_j = \mathbb{E}[\text{dist}(m_j(x), f_k(x))]$. Verifier applies $C_j(x, m_j(x))$ and returns pass if $P(C_j \mid \text{correct}) \geq \tau$. Expected end-to-end error reduction when using macros: if macros are applied with reuse frequency r and verifier true-positive pass rate p, approximate error reduction $\Delta \approx r \cdot (\varepsilon_{\text{raw}} - \varepsilon_j) \cdot p$, where ε_{raw} is average raw sub-chain error without macro reuse.

(C) High-Level Conceptual Workflow

1. Trace collection phase (offline or continual): Collect many generator traces T_i for training problems or live tasks; annotate subchain boundaries by structural cues or automatic segmentation (e.g., argument/operation boundaries).
2. Subchain clustering and macro synthesis: Cluster similar subchains by semantic signature (input/output shape, predicates) and distill each cluster into a macro m_j with a compact parameterization and a succinct interface I_j.
3. Verifier construction: For each macro, design a fast verifier C_j that checks one or more invariants (e.g., arithmetic equality, type constraints, unit consistency, symbolic simplification, or shallow model prediction). Associate a verifier confidence function conf_j.
4. Deployment (inference): For a new query, G composes a reasoning plan. When a subgoal matches a macro signature, invoke m_j instead of expanding the full sub-chain; run C_j on the macro's output. If C_j passes with $\text{conf}_j \geq \tau$, accept and continue; if it fails or is low-confidence, fallback to re-synthesizing the sub-chain with G (optionally producing a repaired macro example).
5. Continuous update: Failed or borderline applications are used to refine m_j (adjust θ_j) and to improve C_j thresholds; successful reuse increments reuse counters for supervisor ranking.

(D) Potential Innovation Directions

1. Formalized Macro Contracts: Define formal pre/post-conditions (contracts) for macros and synthesize verifiers C_j with SMT/symbolic engines for domains where symbolic checks are tractable (e.g., algebra, type systems).
2. Cross-Domain Macro Transfer: Investigate meta-features for macro signature representation enabling reuse across tasks and modalities (textual proofs $\leftrightarrow$ code transformations $\leftrightarrow$ symbolic math).
3. Resource-Aware Macro Selection: Add an energy/latency cost model that chooses between macro invocation and raw re-synthesis to optimize a cost-accuracy objective under latency budgets.
4. On-Device Macro Lifecycles: Enable light-weight on-device macro discovery and caching with privacy-preserving aggregation across devices (federated macro statistics).
5. Verifier Co-training: Co-train verifier V with generator G to improve verifier calibration and enable provable probabilistic guarantees (e.g., PAC-style bounds on macro error under verification).

(E) Theoretical Experimental Design

Objective: Validate that DMGV (dynamic macro discovery + verifier-guided reuse) improves multi-step reasoning accuracy, reduces token/computation usage, and improves calibration relative to standard chain-of-thought (CoT) and self-consistency baselines.

Experimental setup

- **Generator backbone:** a reasonably capable open LLM (e.g., LLaMA-2 13B or equivalent) used for generation and macro synthesis. Use same checkpoint for all methods to ensure fairness.
- **Verifier model:** a lightweight transformer (e.g., 200M–700M) fine-tuned for verification tasks per domain, plus domain-specific symbolic checkers where available.
- **Macro encoder/parameterizer:** a compact adapter (10M–50M params) which maps macro inputs to outputs.
- **Compute budget:** fix average allowed tokens-per-query and wall-clock time budget to model deployment constraints.
- **Datasets (unseen evaluation sets):** use held-out splits from multi-step reasoning benchmarks: GSM8K (arithmetic), ProofWriter (formal reasoning), StrategyQA (multi-hop), and a code-synthesis multi-step reasoning subset (e.g., problem-to-algorithm translation tasks). Use separate corpus for macro discovery (training/validation) and unseen tasks for testing to measure generalization.

Independent variables

- **Method:**
  - Baseline A: CoT prompting (chain-of-thought) with standard temperature (0.7).
  - Baseline B: CoT + self-consistency (N samples, majority voting).
  - Variant C: DMGV-static — macros synthesized offline once from training traces, used at inference without further refinement; verifier present.
  - Variant D: DMGV-dynamic — macros synthesized and updated online with verifier-driven repair (full proposed method).
- **Verifier fidelity (for DMGV experiments):**
  - Low: verifier with conservative pass threshold τ_{low}.
  - High: verifier with higher threshold τ_{high}.
- **Macro reuse budget $r_{\max}$:** maximum macros allowed per query (tune values: 0, 1, 3, 10).

Dependent variables / measurable indicators

- **Final answer accuracy (primary):** fraction of correct final answers on each dataset.
- **Token consumption per query:** average number of model tokens generated (generator + macro expansion).
- **Wall-clock inference latency per query.**
- **Chain length reduction:** average number of low-level reasoning steps synthesized vs baseline.
- **Verification pass rate:** fraction of macro applications accepted without fallback.
- **Calibration/Brier score:** calibration of model confidence on final answers.
- **Error localization:** proportion of failures localized to specific macro applications (for analysis).
- **Computational cost:** compute flops or approximate cost per query.
- **Statistical significance:** use paired bootstrap resampling to test differences in accuracy.

Procedures

- **Macro discovery:** Use a held-out set of training problems to produce traces via G. Apply the DMGV clustering and synthesize macros m_j. For DMGV-dynamic, allow online updates during evaluation by feeding back failed macro instances; for DMGV-static, freeze library once.
- **Baseline runs:** Run Baseline A and B over evaluation sets, collect metrics (accuracy, tokens, latency).
- **DMGV runs:** Evaluate DMGV-static and DMGV-dynamic under multiple verifier fidelity settings and reuse budgets. Ensure identical inference temperature and beam settings for generator across runs.
- **Ablations:**
  - No-verifier ablation: DMGV with macros but no verifier (accept all macros).
  - No-compression ablation: use discovered subchains but do not compress; instead, cache and replay full subchains.
  - Vary $r_{\max}$: measure trade-offs.
- **Statistical analysis:** For each metric, compute mean $\pm$ 95% CI across queries. Use paired statistical tests comparing DMGV-dynamic to best baseline (likely CoT+self-consistency) on accuracy and token savings.

Expected measurable effects (testable hypotheses)

- **H1 (accuracy):** DMGV-dynamic increases final-answer accuracy over Baseline B by an absolute margin $\Delta_{\text{acc}} \geq 5$–$15\%$ on deep multi-step tasks (ProofWriter, GSM8K) due to reduced error accumulation in reused macros.
- **H2 (efficiency):** DMGV reduces average tokens per query by at least 20% when macros are frequently reused ($r \geq 3$) versus Baseline A, while maintaining or improving accuracy.
- **H3 (calibration):** DMGV with verifier improves calibration (lower Brier score) because verification provides meaningful pass/fail signals that correlate with correctness.
- **H4 (targeted repair):** Ablations will show that verifier presence is critical: no-verifier DMGV shows token reduction but loses accuracy, demonstrating verifier's key role in maintaining correctness.

Success criteria to support patent-claim-level benefit

- Statistically significant improvement in accuracy ($p < 0.05$ paired bootstrap) versus CoT+self-consistency on at least two of the four datasets.
- Measurable token/latency savings while meeting or exceeding baseline accuracy in deployed-cost budget scenarios.
- Demonstration that verifier-driven online macro refinement reduces subsequent per-macro error rate ε_j by a measurable fraction (e.g., 30–60%) across update cycles.

This experimental design ties the core invention — validated, compressed reasoning macros with selective verification and repair — to concrete, measurable improvements in accuracy, efficiency, and calibration, and includes ablations to isolate the contribution of discovery, compression, and verification.

Table 12: Demonstration of an idea generated by FlowPIE in the Health & Medicine domain.

Domain: *Health & Medicine*
Final Idea Title: *A Tumor-Core–Activated Therapeutic Strategy* **Summary**: The proposed strategy introduces a tumor-core–activated neutrophil reprogramming blockade that selectively prevents infiltrating neutrophils from converging into the **pro-angiogenic dcTRAIL-R1$^+$ T3 state** identified in the target paper. Rather than depleting neutrophils systemically or blocking VEGF broadly, this approach acts upstream at the level of fate determination, using the hypoxic–glycolytic tumor niche as a spatial trigger to locally release an inhibitor against a critical epigenetic or transcriptional effector of the T3 program. The central advantage is that it suppresses tumor-promoting neutrophil plasticity with spatial precision while preserving peripheral neutrophil host-defense functions, thereby overcoming the major limitations of pan-neutrophil depletion and conventional anti-angiogenic therapy. Conceptually, the strategy combines *(1) microenvironment-selective activation*, ensuring that the payload is engaged primarily within the tumor core; *(2) neutrophil fate interception*, blocking the chromatin remodeling and transcriptional induction required for dcTRAIL-R1 and VEGFα upregulation; and *(3) functional antitumor consequence*, reducing T3 accumulation, intratumoral angiogenesis, and tumor growth. By targeting the newly defined deterministic reprogramming axis rather than only its downstream effector output, this framework offers a more mechanism-based, selective, and potentially safer therapeutic paradigm for modulating tumor-associated neutrophils in solid cancers.

Table 13: Demonstration of an idea generated by FlowPIE in the Genetics & Molecular Biology domain.

Domain: *Genetics & Molecular Biology*
Final Idea Title: *A system-and-method for "sticker-resolved flow-activation spectroscopy" of biomolecular condensates* **Summary**: The proposed framework introduces sticker-resolved flow-activation spectroscopy (SRFAS), a combined experimental–computational system that quantitatively separates the **energetic barrier governing condensate network rearrangement** from the **chain-length–dependent kinetic prefactors that control relaxation dynamics**. Instead of relying on conventional rheology that conflates binding energetics with polymer architecture, SRFAS integrates temperature-dependent microrheology, sequence-tunable transient probe tracers, and Bayesian inversion of coarse-grained transient-network models to independently infer two key parameters: the flow activation energy (E_a), reflecting sticker binding free energies, and the kinetic prefactor $(\tau_0(N, v, c))$, reflecting chain length, valence, and network topology. This separation enables direct recovery of molecular-scale interaction energetics from purely bulk and tracer-level rheological observables, providing predictive control over macroscopic properties such as viscosity, relaxation time, and molecular diffusion in biomolecular condensates. By coupling experimentally measurable viscoelastic spectra, tracer diffusion, and residence-time distributions with model-based inference, the system delivers experimentally verifiable, sequence-to-rheology mapping that surpasses conventional microrheological approaches in both parameter identifiability and predictive accuracy. As a result, SRFAS establishes a general platform for rational design, screening, and mechanistic analysis of biomolecular condensates and associative polymer materials, with applications ranging from understanding disease-related condensate mutations to engineering synthetic phase-separating biomaterials with programmable dynamic properties.

Table 14: Demonstration of an idea generated by FlowPIE in the Environmental Sciences domain.

Domain: *Environmental Sciences*
Final Idea Title: *A transferable Diagnostic-and-Prediction system for ecosystem functional loss under rare short-term droughts* **Summary**: The proposed DP-Extreme (Diagnostic-and-Prediction system for Extreme drought impacts) introduces a transferable framework that quantitatively predicts ecosystem productivity loss under **rare short-term drought events** and guides optimized mitigation decisions. Unlike conventional precipitation–productivity models that assume linear sensitivity to rainfall deficits, DP-Extreme explicitly incorporates drought rarity (return period), pre-drought resource stocks (soil moisture and nutrients), and community resilience traits (biodiversity and rooting depth) into a hierarchical scaling law that captures the nonlinear sensitivity of aboveground net primary production (ANPP) to extreme pulses. By combining standardized pulse-exclusion experiments across environmental gradients with Bayesian parameter inference, the system estimates site-specific parameters that generate an **Extreme Sensitivity Index (ESI)**—a predictive indicator of expected ANPP loss for rare drought scenarios (e.g., 1-in-100-year events). This enables threshold-triggered mitigation strategies such as targeted irrigation or grazing control that activate only when predicted loss exceeds critical levels. The key advantage is that DP-Extreme decouples drought rarity effects from ecological buffering traits, producing substantially more accurate forecasts of ecosystem productivity loss and enabling resource-efficient interventions. In validation experiments, the system is designed to demonstrate significantly lower prediction error and improved decision efficiency compared with conventional climate-only or linear sensitivity models, establishing a practical decision-support platform for managing ecosystem resilience under increasingly frequent extreme climate events.