HypoForge: A Self-Improving Multi-Agent Framework for Automated Hypothesis Generation and Testing via Scientific Skill Learning

Ziqing Qian[1,2], **Jiaying Lei**[1,2,3], **Yifang Wang**[4], **Nan Cao**[1,2,3]*
[1]Intelligent Big Data Visualization Lab, Tongji University, Shanghai, China
[2]Shanghai Research Institute for Intelligent Autonomous System, Tongji University, Shanghai, China
[3]Shanghai Innovation Institute, Shanghai, China
[4]Florida State University Tallahassee, Florida, USA
2411920@tongji.edu.cn, jiaying.lei@outlook.com, yifang.wang@fsu.edu, nan.cao@gmail.com

Abstract

Large language models (LLMs) have enabled AI scientist systems to automate scientific discovery, yet existing approaches most rely on static prompting or fixed workflows and fail to accumulate experience for continual improvement. We propose ***HypoForge***, an experience-guided multi-agent framework that learns reusable scientific skills for automated hypothesis generation and hypothesis testing. *HypoForge* is built on the observation that these two stages involve different supervision signals. For hypothesis generation, where explicit feedback is unavailable, *HypoForge* adopts an adversarial generator–discriminator mechanism to improve reasoning through comparative critique. For hypothesis testing, where empirical feedback is available, *HypoForge* learns testing skills from execution outcomes and ground-truth results. By matching skill learning strategies with stage-specific supervision, *HypoForge* enables continual improvement without fine-tuning foundation models. Experiments on hypothesis generation and testing benchmarks show that *HypoForge* consistently outperforms existing AI scientist frameworks and skill-level variants. Further analysis demonstrates the effectiveness of the proposed stage-specific skill learning paradigms.

Introduction

Hypothesis generation and hypothesis testing constitute the core of hypothesis-driven scientific research. Given a research question and empirical observations, scientists formulate candidate hypotheses to explain underlying phenomena and subsequently validate them through statistical analyses and experimental evidence. More importantly, scientific research is inherently cumulative: researchers continuously summarize reusable scientific skills from previous successes and failures, enabling them to generate better hypotheses, design more reliable experiments, and solve increasingly complex scientific problems.

Recent advances in large language models (LLMs) have led to the emergence of AI scientist systems capable of automating scientific reasoning (Wang et al. 2023b), hypothesis generation (Zhou et al. 2024), experiment design (Boiko et al. 2023), code generation (Yang et al. 2024), and data analysis (Gu et al. 2024). Although these systems have demonstrated impressive reasoning capabilities, most either focus on a single stage of scientific discovery, producing hypotheses that are never tested against data or testing hypotheses that must be supplied externally, or operate through static prompting or predefined workflows, limiting their ability to distill, accumulate, and reuse scientific experience across tasks. As a result, they repeatedly pay similar reasoning efforts from scratch across tasks, leading to reduced discovery efficiency (Wang et al. 2023a).

We therefore ask how the discovery experience can be distilled into reusable scientific skills across the two stages that constitute the discovery loop: hypothesis generation and hypothesis testing. The central challenge is that the two stages differ fundamentally in the supervision available for skill learning. Hypothesis generation is an open-ended scientific reasoning task that lacks explicit supervision, making it difficult to directly assess hypothesis quality. Consequently, improvement relies on comparing and critiquing alternative hypotheses to gradually distill effective reasoning strategies. In contrast, hypothesis testing naturally provides empirical supervision through executable experiments and ground-truth outcomes, enabling direct evaluation of experimental design and implementation. These fundamentally different supervision signals imply that hypothesis generation and hypothesis testing require stage-specific learning paradigms rather than a unified refinement strategy.

To address this, we propose ***HypoForge***, a self-improving multi-agent framework for automated hypothesis generation and hypothesis testing. Rather than simply storing previous reasoning trajectories, *HypoForge* continuously distills accumulated experience into reusable scientific skills that improve future reasoning. Specifically, for hypothesis generation, *HypoForge* introduces an adversarial generator–discriminator framework that iteratively learns hypothesis-generation skills through comparative critique without explicit supervision. For hypothesis testing, HypoForge exploits execution outcomes and ground-truth feedback to iteratively learn testing skills under empirical supervision. By aligning skill learning with the supervision characteristics of each stage, *HypoForge* enables continual capability improvement without requiring fine-tuning of the underlying foundation models. We evaluate *HypoForge* on benchmark datasets for automated hypothesis generation and hypothesis testing. Experimental results demonstrate that *HypoForge* consistently outperforms existing AI scientist frameworks

*Corresponding author.

and skill-level variants in both hypothesis quality and testing performance. Extensive ablation studies further verify the effectiveness of the proposed stage-specific skill learning paradigms. Our contributions are summarized as follows:

- We propose *HypoForge*, a self-improving multi-agent framework that learns reusable scientific skills for automated hypothesis generation and hypothesis testing.
- We identify the distinct supervision characteristics of hypothesis generation and hypothesis testing, and develop two stage-specific skill learning paradigms: an adversarial self-improvement mechanism for hypothesis generation without explicit supervision, and a ground-truth feedback learning mechanism for hypothesis testing with empirical supervision.
- Extensive experiments demonstrate that *HypoForge* consistently improves both hypothesis generation and hypothesis testing over existing AI scientist frameworks, while the learned skills exhibit strong transferability across diverse scientific research tasks.

Related Work

In this section, we review two research directions most related to our work: autonomous scientific discovery and skill learning for LLM agents.

Autonomous Scientific Discovery

Autonomous scientific discovery aims to enable machines to generate and validate novel scientific hypotheses from empirical evidence (Wang et al. 2023b). Central to autonomous scientific discovery are hypothesis generation and hypothesis testing, which form an iterative reasoning loop that transforms empirical observations into testable explanations and evaluates whether those explanations are supported by evidence. With recent advances in artificial intelligence (AI) and large-scale data-driven approaches, automated hypothesis generation and testing have attracted increasing attention. Takagi et al. demonstrated the potential of LLMs for scientific ideation through knowledge synthesis (Takagi, Yamauchi, and Kumagai 2023), while Zhou et al. showed that prompting strategies and contextual information significantly affect generated hypotheses (Zhou et al. 2024). Recent benchmarks, including HypoBench and DiscoveryBench, have established standardized evaluations for LLM-based hypothesis generation (Liu et al. 2025; Majumder et al. 2025). Beyond generation tasks, a few studies have also explored automated hypothesis testing by integrating LLM reasoning with computational experiments and iterative testing, including self-improving scientific agents (Brunnsåker et al. 2025), sequential falsification frameworks (Huang et al. 2025), and multi-agent discovery workflows (Gupta et al. 2026).

Despite these advances, existing systems typically focus on a single stage of scientific discovery, either hypothesis generation or hypothesis testing, and rely on the pretrained model's fixed reasoning ability to solve each new task, thus limiting their ability to accumulate experience and progressively improve their scientific reasoning. Our work bridges this gap through agent skill learning, integrating hypothesis generation and testing into a unified discovery process while distilling accumulated discovery experiences into reusable reasoning capabilities.

Agent Skill Learning

Recent LLM agents have evolved from stateless reasoning toward experience-driven capability improvement. Memory-augmented agents, such as Generative Agents and MemGPT, introduced mechanisms to store and retrieve long-term information to support future interactions (Park et al. 2023; Packer et al. 2023). Reflexion further enable agents to learn from past failures (Shinn et al. 2023), while Self-Refine demonstrated iterative refinement through self-generated feedback without updating model parameters (Madaan et al. 2023). Building upon these ideas, a few studies have investigated agent skill learning as a way to distill reusable capabilities from accumulated past experiences. Voyager pioneered automatic skill library construction, allowing agents to discover, store, and reuse executable skills during open-ended interaction (Wang et al. 2023a). Subsequent research has extended this paradigm from skill storage toward skill acquisition, optimization, and refinement. For example, reinforcement learning-based approaches explore how agents can improve skill libraries through interaction experiences (Wang et al. 2026), while another line of research investigates automatic skill optimization and revision through trajectory feedback and testing approaches (Yang et al. 2026; Liu et al. 2026). More recent studies further explore self-evolving skill frameworks that improve agent capabilities through iterative skill generation, evaluation, and refinement(Zhang et al. 2026; Vishe et al. 2026).

However, existing skill-learning approaches typically optimize agent skills in a unified manner, aggregating experiences and feedback from different stages without explicitly modeling their distinct capabilities. This limits stage-specific skill refinement and feedback attribution, making it difficult to determine how accumulated experiences should be transformed into targeted procedural skills for complex multi-step workflows. In this work, we address this problem by designing a stage-specific skill learning framework that leverages different feedback to distill transferable strategies for hypothesis generation and testing.

Problem Formulation

Given a user research problem P and an associated data source D, the multi-agent system aims to generate a set of hypotheses $\mathcal{H} = \{h_i\}_{i=1}^{M}$ and corresponding testing trajectories $\mathcal{V} = \{\tau_i\}_{i=1}^{M}$, where $\tau_i = (h_i, e_i, c_i, r_i)$ denotes the testing trajectory of each hypothesis, including the hypothesis (h_i), experimental design (e_i), executable code (c_i), and execution outcome (r_i). For each iteration t, the system generates hypotheses $\mathcal{H}^t$ and testing trajectories $\mathcal{V}^t$ based on the current skills and accumulated discovery experiences. The objective is to progressively learn three reusable procedural agent skills $\mathcal{S}^t = \{\mathcal{S}_h^t, \mathcal{S}_e^t, \mathcal{S}_x^t\}$ for hypothesis generation, experimental design, and experiment execution, respectively. The skills are iteratively updated by the multi-agent system based on newly acquired experiences and feedback from

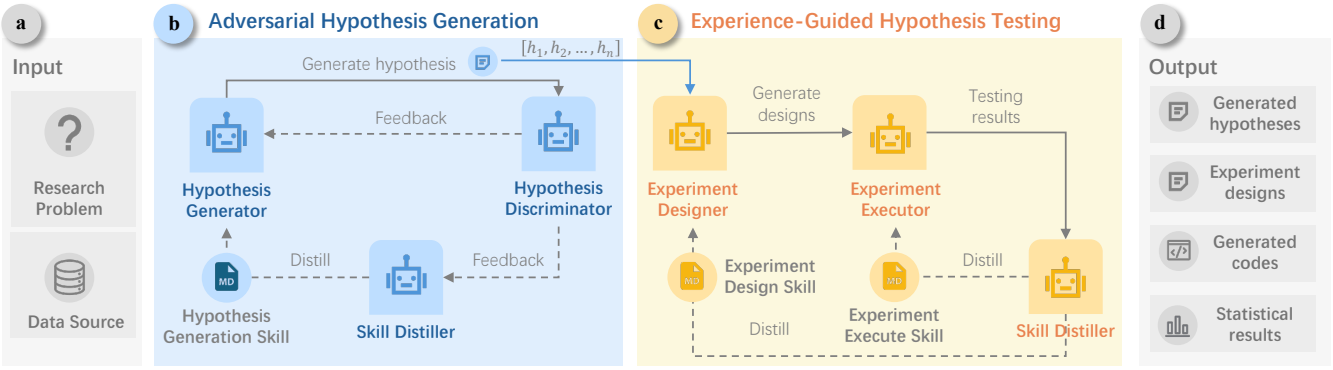


Figure 1: Multi-agent framework that supports hypothesis generation and testing with skill learning.

training records:

$$\mathcal{S}_h^{t+1} = \Phi_h(\mathcal{S}_h^t, \mathcal{H}^t, \mathcal{F}_h^t), \tag{1}$$

$$\mathcal{S}_e^{t+1}, \mathcal{S}_x^{t+1} = \Phi_t(\mathcal{S}_e^t, \mathcal{S}_x^t, \mathcal{V}^t), \tag{2}$$

where Φ_h denote the skill learning processes for hypothesis generation, $\mathcal{F}_h^t$ denotes the feedback obtained from hypothesis evaluation, and Φ_t for hypothesis testing.

Method

In this section, we present *HypoForge*, a stage-wise skill learning framework for scientific hypothesis generation and testing. As illustrated in Fig. 1, the framework decomposes hypothesis-driven scientific research process into two stages and learns dedicated skills for each stage according to its distinct reasoning process and supervision signals. The adversarial hypothesis generation stage (Fig. 1(b)) iteratively refines hypothesis construction through interactions between a hypothesis generator and a discriminator, distilling multi-dimensional evaluation feedback into reusable hypothesis-generation skills. The experience-guided hypothesis testing stage (Fig. 1(c)) learns reusable testing skills from prior testing trajectories by exploiting ground-truth outcomes and feedback from experiment design, execution, and validation.

Adversarial Hypothesis Generation

Inspired by adversarial learning in Generative Adversarial Networks (GANs) (Goodfellow et al. 2014), we formulate hypothesis generation as an adversarial skill learning problem (Fig. 1(b)). Specifically, a hypothesis generator produces a batch of candidate hypotheses that approximates the distribution of plausible scientific discoveries, while a multi-dimensional discriminator evaluates the generated hypothesis set as a whole. Unlike the binary discriminator in conventional GANs, our discriminator is formulated as a scoring function that measures how closely the generated hypothesis distribution matches the characteristics of high-quality human-written scientific hypotheses. The resulting distribution-level feedback is distilled into reusable hypothesis-generation skills, enabling the generator to progressively improve its generation policy across iterative interactions. Compared with Reviewer/Critic-based multi-agent frameworks, which iteratively revise individual generation items using instance-level feedback, our framework performs distribution-level optimization over the entire hypothesis set. Consequently, the learned skills capture transferable generation strategies, including variable selection, causal reasoning, hypothesis construction, and avoidance of common hypothesis generation failures, rather than hypothesis-specific corrections, leading to better generalization across different hypothesis-driven scientific research tasks.

Hypothesis Generator Given a research problem P and data source D, the hypothesis generator G_h produces a batch of candidate hypotheses,

$$\mathcal{H}^t = \{h_1^t, h_2^t, \ldots, h_N^t\} = G_h(P, D, \mathcal{S}_h^t), \tag{3}$$

where $\mathcal{S}_h^t$ denotes the hypothesis generation skill accumulated from previous adversarial interactions.

The generated hypothesis set provides a distributional approximation of plausible scientific discoveries, enabling the discriminator to jointly evaluate the scientific quality, diversity, and discovery coverage of the generated hypotheses. Each hypothesis is represented as a structured scientific claim describing a testable relationship between variables together with its expected direction. The generator prompt specifies only the agent role, research objective, output format, and general constraints, while the learned skill $\mathcal{S}_h^t$ encapsulates reusable procedural knowledge distilled from previous adversarial interactions, including informative variable selection, causal reasoning patterns, hypothesis construction strategies, and common failure avoidance.

Multi-Dimensional Hypothesis Discriminator The proposed discriminator D_h is formulated as a multi-dimensional scoring function rather than a binary classifier. Instead of determining whether a generated hypothesis is "real" or "fake", D_h estimates how closely the generated hypothesis set matches the latent distribution of high-quality human-written scientific hypotheses.

High-quality scientific hypotheses generally satisfy several fundamental principles established in the scientific discovery literature, including empirical grounding (King et al. 2009; Langley 1987), causal validity (Neuberg 2003), falsifiability (Popper 2005), theoretical consistency (Lakatos 1978),

clarity and operationalizability (Shapere 1964), and scientific novelty (Fortunato et al. 2018). These principles are summarized into seven evaluation dimensions:

$$\mathbf{d}_i = [D_1^i, D_2^i, \ldots, D_7^i], \tag{4}$$

where $D_1, D_2, D_5, D_6 \in [0, 1]$ measure empirical grounding, causal plausibility, structural clarity, and scientific novelty, respectively, while $D_3, D_4, D_7 \in \{0, 1\}$ serve as hard constraints enforcing falsifiability, theoretical consistency, and empirical verifiability. Collectively, these measurements (implemented based LLM detailed in the supplement materials) characterize the essential properties shared by high-quality human-written scientific hypotheses as follows:

$$s_i = D_3^i D_4^i D_7^i \left(D_1^i D_2^i D_5^i D_6^i\right)^{\frac{1}{4}}, \tag{5}$$

where larger values indicate greater consistency with the characteristics of high-quality human-written hypotheses. The overall scientific quality of the generated hypothesis set is computed as:

$$Q(\mathcal{H}^t) = \frac{1}{N} \sum_{i=1}^{N} s_i. \tag{6}$$

Besides evaluating hypothesis quality, the discriminator further compares the generated hypothesis set with reference hypotheses to provide reflective feedback:

$$C(\mathcal{H}^t) = Match(\mathcal{H}^t, \mathcal{H}_{gt}), \tag{7}$$

where $\mathcal{H}_{gt}$ denotes the reference hypothesis set. Unlike hypothesis quality, this comparison provides additional insights into the discovery patterns covered by the generated hypotheses. Specifically, hypotheses with high overlap with reference hypotheses reveal effective scientific patterns, while hypotheses with limited overlap indicate potential missing factors or unexplored directions for future exploration.

Adversarial Skill Learning Inspired by adversarial learning, the generator progressively improves its hypothesis generation policy by maximizing the scientific quality score,

$$\max_{G_h} \mathcal{L}_G = Q(\mathcal{H}^t). \tag{8}$$

Unlike conventional GANs, the discriminator is not optimized through binary classification, nor is the generator updated via gradient back-propagation. Instead, the discriminator summarizes the strengths and weaknesses of the generated hypothesis set and distills them into reusable procedural knowledge:

$$\mathcal{S}_h^{t+1} = Distill\left(\mathcal{S}_h^t, \mathcal{H}^t, Q(\mathcal{H}^t), C(\mathcal{H}^t)\right), \tag{9}$$

where $Distill(\cdot)$ is a LLM-based agent that extracts transferable hypothesis-generation strategies from discriminator feedback and accumulated hypothesis experiences.

The updated skill $\mathcal{S}_h^{t+1}$ is incorporated into the generator in the next iteration, enabling continuous improvement of the generation policy. By distilling distribution-level feedback into reusable procedural knowledge, the generator progressively aligns with the distribution of high-quality human-written hypotheses, achieving stronger generalization across hypothesis-driven scientific research tasks.

Experience-Guided Hypothesis Testing

Hypothesis testing systematically evaluates candidate scientific hypotheses by transforming them into executable empirical studies and determining whether they are supported by observational evidence. Given a hypothesis, the testing process consists of designing an experiment protocol, implementing the corresponding testing procedure, executing the analysis on the target dataset, and interpreting the resulting evidence. Unlike hypothesis generation, hypothesis testing provides explicit supervision through ground-truth outcomes, allowing the system to evaluate the quality of the entire testing process. Building upon this property, we propose an experience-guided hypothesis testing framework that continuously improves experiment design and execution through iterative skill learning. Specifically, the framework first generates a textual experiment protocol for each hypothesis, then implements and executes the corresponding testing program, and finally compares the obtained testing results with the ground truth to analyze the complete testing trajectory. The resulting recommendations are distilled into reusable experiment design and execution skills, enabling progressively more rigorous and reliable hypothesis testing. Specifically, the experiment design skill accumulates methodological experience, such as data variable selection, confounder control, and analysis method selection, for constructing reliable experiment plans. The execution skill distills engineering experience to realize these experiment plans in code, including code generation, library usage, debugging, and reliable execution of statistical procedures.

Experiment Design Given a hypothesis h_i, the experiment designer agent G_e constructs a textual experiment protocol describing how the hypothesis should be empirically tested on the target dataset D:

$$e_i = G_e(h_i, D, \mathcal{S}_e^t), \tag{10}$$

where $\mathcal{S}_e^t$ denotes the learned experiment design skill. The protocol e_i serves as a high-level testing specification that defines all essential components of the experiment, including hypothesis operationalization, variable selection, data preprocessing and filtering, confounder control, statistical analysis methods, significance criteria, evaluation metrics, and the complete experimental workflow. Rather than encoding task-specific instructions, the design skill $\mathcal{S}_e^t$ provides reusable procedural knowledge distilled from previous testing experiences, enabling the agent to progressively construct more rigorous and scientifically sound experiment protocols.

Experiment Execution Given e_i, the execution agent G_x generates an executable testing program c_i as follows:

$$c_i = G_x(e_i, \mathcal{S}_x^t), \tag{11}$$

where $\mathcal{S}_x^t$ denotes the learned execution skill. c_i is subsequently executed on the target dataset:

$$r_i = Exec(c_i, D), \tag{12}$$

where r_i contains the complete testing outcomes, including statistical results, significance tests, estimated effect sizes, and the final testing conclusion.

The execution skill $\mathcal{S}_x^t$ captures reusable implementation knowledge for translating e_i into reliable programs c_i, including statistical implementation, code organization, execution robustness, exception handling, debugging strategies, and standardized result reporting. The complete testing trajectory is then represented as:

$$\tau_i = (h_i, e_i, c_i, r_i), \tag{13}$$

which records the hypothesis h_i, experiment protocol e_i, implementation c_i, and testing outcomes r_i for subsequent skill refinement.

Experience-Based Skill Refinement After executing the testing procedure, the framework compares the obtained testing outcome with the corresponding ground-truth result to evaluate the complete testing trajectory. The ground-truth outcome provides explicit supervision for evaluating both the testing conclusion and the validity of the underlying experiment design and implementation.

Specifically, given a testing trajectory $\tau_i = (h_i, e_i, c_i, r_i)$, the framework optimizes the experiment design and execution skills according to the test pass rate:

$$(\mathcal{S}_e^*, \mathcal{S}_x^*) = \arg\max_{\mathcal{S}_e, \mathcal{S}_x} T_h, \tag{14}$$

where T_h denotes the proportion of ground-truth hypotheses that are successfully tested by the generated experiments. Through iterative skill refinement, the framework aims to improve the reliability of experiment design and execution for future hypothesis testing.

The accumulated testing trajectories and corresponding ground-truth outcomes are subsequently provided to the skill distiller for experience-based refinement:

$$(\mathcal{S}_e^{t+1}, \mathcal{S}_x^{t+1}) = Distill\left(\mathcal{S}_e^t, \mathcal{S}_x^t, \{\tau_i\}_{i=1}^N\right), \tag{15}$$

where $Distill(\cdot)$ is an LLM-based agent that analyzes previous testing trajectories and execution outcomes, attributes testing successes and failures to different stages of the experimental process, and extracts transferable experimental practices rather than memorizing individual cases.

The refined skills are incorporated into the experiment designer and execution agents in subsequent iterations, enabling stage-wise improvement of the entire hypothesis testing pipeline. By continuously refining experiment design and execution skills through accumulated testing experiences, the proposed framework acquires transferable testing capabilities that generalize across diverse hypothesis-driven scientific research tasks.

Experiments

In this section, we conduct comprehensive experiments to evaluate the effectiveness of our proposed framework. We begin by describing the experimental settings. Then, we compare our framework with baselines on both hypothesis generation and hypothesis testing tasks. Finally, we perform detailed skill learning analysis to investigate skill evolution during refinement, and conduct ablation studies to evaluate the role of different refinement signals.

Experimental Setup

We first introduce the experimental setup, including the benchmark datasets, comparison baselines, evaluation metrics, and implementation details.

Datasets We evaluate our framework on **HypoBench** (Liu et al. 2025), a benchmark for scientific hypothesis generation and discovery. The original benchmark contains 13 tasks from diverse domains, and we use all tasks to construct our experimental suite. The detailed task adaptation and preprocessing procedures are provided in the supplement materials, and the statistics of all tasks are summarized in Table 1. Each task consists of a research question, an associated dataset, and a set of reference hypotheses. For the hypothesis generation stage, we use representative data from the selected 13 tasks. For the hypothesis testing stage, we retain only tasks with empirically evaluable ground-truth hypotheses, as trajectory-level supervision from testing outcomes is required to refine experiment design and execution skills.

Task	Type	#Data Size
Deceptive Reviews Detection	Real	800
Dreddit Mental Stress Detection	Real	200
AI-Generated Content Detection	Real	200
News Headline Engagement	Real	200
Persuasive Argument Prediction	Real	200
Retweet Prediction	Real	200
Paper Citations (Health)	Real	244
Paper Citations (NeurIPS)	Real	308
Paper Citations (Radiology)	Real	392
Shoe Sales	Synthetic	900
College Admission	Synthetic	200
Presidential Election	Synthetic	1,750
Personality Prediction	Synthetic	1,750

Table 1: HypoBench tasks used in our experiments.

Baselines We compare *HypoForge* against two types of baselines: system-level scientific discovery approaches and skill-level variants.

- **System-level comparison.** For hypothesis generation, we use HypoGeniC (Zhou et al. 2024), which provides an automated pipeline for hypothesis generation, and HypotheSAEs (Movva et al. 2025), which derives hypotheses by interpreting sparse autoencoder features from neural representations. For hypothesis testing, we include POPPER (Huang et al. 2025), a scientific discovery framework that supports automated hypothesis validation, and ReAct (Yao et al. 2022), which enables LLM agents to iteratively reason, plan, and execute actions for experiment design and validation. In addition, we include LLM-ZeroShot and LLM-FewShot (Brown et al. 2020) as general LLM-based baselines for both tasks.
- **Skill-level comparison.** We include three skill variants to compare the effectiveness of learned skills on both tasks. (1) No-Skill: removes the learned skill and uses only the original agent framework; (2) AI-Generated-Skill: replaces the learned skill with an LLM-generated procedu-

ral skill, without refinement through task-specific experiences or evaluation feedback; (3) Human-Designed-Skill: replaces the learned skill with a pre-defined manually specified skill without refinement. (K-Dense Inc. 2026)

Metrics For hypothesis generation, we use $Q(\mathcal{H}^t)$ score to evaluate the quality of generated hypotheses, and $Hit@K$ to measure the coverage of generated hypotheses against reference hypotheses. For hypothesis testing, we evaluate the effectiveness of automatically generated experiments using Test Pass Rate T_h and Execution Success Rate E_h. T_h measures the proportion of ground-truth hypotheses that are successfully validated by the generated experiments, where successful validation requires the experiment to execute successfully and achieve statistical significance ($p < 0.05$). E_h measures the proportion of experiments that successfully execute and produce valid results.

Implementation Details We divide the selected HypoBench tasks into training and testing sets at a 1:1 ratio, where the training split is used for skill learning and the unseen test split is reserved for evaluation. All agents are instantiated with DeepSeek-V4-Flash (DeepSeek-AI 2026) as the backbone model. During training, skill updates are retained only when the corresponding learning objective improves; otherwise, the previous skill is preserved for subsequent iterations, and the training process terminates when no improvement is observed for k consecutive attempts ($k = 3$). All reported results are averaged over three independent runs.

Hypothesis Generation Evaluation

We first evaluate the effectiveness of the hypothesis generation stage on test tasks. All methods receive the same input (research problem and data source pairs), and their generated hypotheses are evaluated using the same discriminator and reference hypotheses. We compare *HypoForge* with both system-level approaches and skill-level variants as mentioned in the above baselines.

Method	$Q(\mathcal{H}^t) \uparrow$	$Hit@K \uparrow$
HypoGeniC	0.719 ± 0.014	0.472 ± 0.050
HypotheSAEs	0.582 ± 0.043	0.208 ± 0.038
LLM-ZeroShot	0.701 ± 0.006	0.516 ± 0.029
LLM-FewShot	0.719 ± 0.007	0.522 ± 0.022
No-Skill	0.724 ± 0.040	0.409 ± 0.079
AI-Generated-Skill	0.738 ± 0.050	0.579 ± 0.022
Human-Designed-Skill	0.761 ± 0.040	0.497 ± 0.022
HypoForge	$\mathbf{0.785 \pm 0.030}$	$\mathbf{0.648 \pm 0.142}$

Table 2: Hypothesis generation performance.

As shown in Table 2, *HypoForge* achieves the best performance among all compared methods, obtaining the highest hypothesis quality score ($Q(\mathcal{H}^t) = 0.785$) and Hit Rate (0.648). Compared with system-level baselines, *HypoForge* improves Hit Rate over HypoGeniC and HypotheSAEs by 17.6% and 44.0%, respectively, demonstrating its ability to generate hypotheses with stronger scientific quality and broader coverage of reference hypotheses. It also consistently outperforms LLM-based prompting baselines, indicating that learned procedural skills provide more effective guidance than static instructions or few-shot demonstrations. For skill-level variants, removing skill learning from our framework decreases the Hit Rate from 0.648 to 0.409, while replacing learned skills with AI-generated or human-designed skills also leads to performance degradation. These results reflect that the proposed adversarial hypothesis generation framework effectively improves the overall performance of the hypothesis generation stage by introducing discriminator-guided refinement.

Hypothesis Testing Evaluation

We then evaluate the hypothesis testing stage on test tasks using available ground-truth hypotheses. All methods are evaluated under the same experimental environment, where generated designs are executed on the provided datasets and assessed according to testing outcomes including experiment designs, experiment codes, and statistical results.

Method	$T_h \uparrow$	$E_h \uparrow$
POPPER	0.417 ± 0.141	$\mathbf{0.984 \pm 0.000}$
ReAct	0.589 ± 0.078	0.859 ± 0.051
LLM-ZeroShot	0.516 ± 0.014	0.870 ± 0.030
LLM-FewShot	0.539 ± 0.008	0.966 ± 0.005
No-Skill	0.562 ± 0.055	0.885 ± 0.099
AI-Generated-Skill	0.555 ± 0.090	0.891 ± 0.131
Human-Designed-Skill	0.612 ± 0.096	0.911 ± 0.043
HypoForge	$\mathbf{0.659 \pm 0.064}$	$\mathbf{0.966 \pm 0.016}$

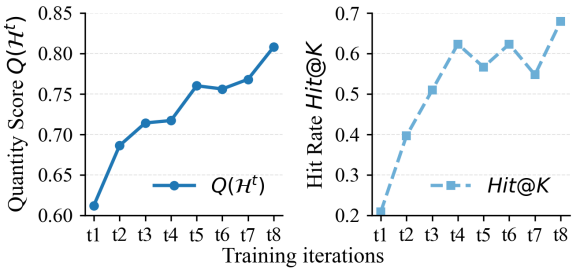
Table 3: Hypothesis testing performance on test tasks.

As shown in Table 3, *HypoForge* achieves the highest test pass rate ($T_h = 0.659$) while maintaining a competitive execution success rate ($E_h = 0.966$). Compared with system-level baselines, *HypoForge* substantially improves testing effectiveness over POPPER, ReAct, and LLM prompting methods, demonstrating the advantage of learning experiment design and execution strategies from previous testing trajectories. Notably, although POPPER achieves a higher execution success rate, its substantially lower test pass rate suggests that many of its generated experiments are technically executable but scientifically ineffective for validating the target hypotheses. Compared with skill-level variants, *HypoForge* consistently outperforms No-Skill, AI-Generated-Skill, and Human-Designed-Skill, confirming that task-specific feedback-driven skill refinement is crucial for improving scientific testing capability.

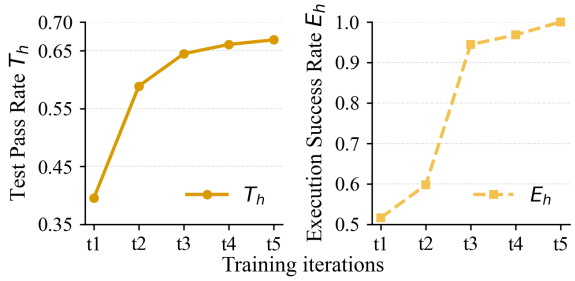
Skill Learning Analysis

To analyze how the proposed skill learning approach improves agent capabilities over iterations, we track the evolution of evaluation metrics during the skill refinement process. Figure 2 shows the learning progress of hypothesis generation and testing skills.

For hypothesis generation, as shown in Figure 2a, both $Q(\mathcal{H}^t)$ and Hit@K exhibit an overall increasing trend across



(a) Hypothesis generation skill learning process.



(b) Hypothesis testing skill learning process.

Figure 2: Skill learning progress across refinement iterations.

refinement iterations. Specifically, $Q(\mathcal{H}^t)$ improves from 0.612 at the initial iteration to 0.808 after skill refinement, while Hit@K increases from 0.208 to 0.679, indicating that the learned skill enables the generator to produce hypotheses with not only higher scientific quality but also broader coverage of reference hypotheses. Although the metrics fluctuate in several intermediate iterations, the overall improvement demonstrates that discriminator feedback provides effective signals for refining hypothesis construction strategies rather than merely selecting high-scoring outputs.

For hypothesis testing, Figure 2b shows consistent improvements in both testing effectiveness and execution correctness. The T_h rate increases from 0.395 to 0.669, while E_h improves from 0.516 to 1.000. This indicates that the learned testing skill progressively captures more effective experiment design and execution procedures.

Ablation Study

To evaluate the impact of refinement signals, we remove the corresponding supervision sources in both stages. For hypothesis generation, w/o Feedback removes discriminator feedback during skill refinement. For hypothesis testing, w/o Outcome removes testing results during skill refinement, while retaining the generated hypotheses, experiment designs, and executable codes.

As shown in Table 4, stage-specific refinement signals consistently improve performance. In hypothesis generation, the full model outperforms w/o Feedback in both $Q(\mathcal{H}^t)$ and $Hit@K$ (0.785 vs. 0.726 and 0.648 vs. 0.491), showing that discriminator feedback effectively refines generation skills. In hypothesis testing, removing execution outcomes decreases T_h from 0.659 to 0.565 while maintaining comparable E_h, demonstrating the importance of empirical validation signals for improving testing skills.

Variant	$Q(\mathcal{H}^t) \uparrow$	$Hit@K \uparrow$
w/o Feedback	0.726 ± 0.008	0.491 ± 0.015
Full	0.785 ± 0.030	0.648 ± 0.142

Variant	$T_h \uparrow$	$E_h \uparrow$
w/o Outcome	0.565 ± 0.084	0.957 ± 0.039
Full	0.659 ± 0.064	0.966 ± 0.016

Table 4: Ablation study of skill learning.

Discussion

Our framework demonstrates the potential of experience-guided skill learning to improve the performance of hypothesis generation and testing without updating the underlying agents and foundation models. Despite these advantages, several limitations remain: (1) the effectiveness of skill refinement currently relies on the quantity and reliability of feedback information. Errors in hypothesis evaluation or insufficient ground-truth validation may lead to suboptimal skill updates. Future work could explore stronger scientific evaluators, multi-agent consensus mechanisms, or human-in-the-loop feedback to improve feedback reliability; (2) the current framework mainly focuses on data-driven scientific discovery with executable computational experiments. Extending skill learning to domains involving complex theoretical reasoning, wet lab, or interdisciplinary knowledge integration remains an important direction; (3)although learned skills may transfer across tasks, automatically determining the granularity, composition, and organization of scientific skills within a complex discovery pipeline remains an open challenge for achieving effective task performance.

Conclusion

In this work, we propose *HypoForge*, an experience-guided multi-agent framework for hypothesis generation and hypothesis testing in the scientific discovery process. Unlike existing AI scientist systems that treat discovery stages as independent tasks and rely on static model capabilities, *HypoForge* enables AI agents to continuously improve their capabilities by learning procedural skills from previous discovery experiences. Specifically, we introduce adversarial refinement for hypothesis generation, where discriminator-guided feedback helps distill effective hypothesis construction strategies without explicit supervision, and experience-guided refinement for hypothesis testing, where empirical validation outcomes improve experiment design and execution skills. Extensive experiments on scientific discovery benchmarks demonstrate that *HypoForge* consistently improves both hypothesis generation and testing performance, while the learned skills generalize across diverse scientific tasks. These results suggest a promising direction in autonomous scientific discovery: developing continually improving AI agent systems that accumulate experience and progressively strengthen their capabilities for scientific reasoning.

References

Boiko, D. A.; MacKnight, R.; Kline, B.; and Gomes, G. 2023. Autonomous chemical research with large language models. *Nature*, 624(7992): 570–578.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Brunnsåker, D.; Gower, A. H.; Naval, P.; Bjurström, E. Y.; Kronström, F.; Tiukova, I. A.; and King, R. D. 2025. Self-driven Biological Discovery through Automated Hypothesis Generation and Experimental Validation. *bioRxiv*, 2025–06.

DeepSeek-AI. 2026. DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence.

Fortunato, S.; Bergstrom, C. T.; Börner, K.; Evans, J. A.; Helbing, D.; Milojević, S.; Petersen, A. M.; Radicchi, F.; Sinatra, R.; Uzzi, B.; et al. 2018. Science of science. *Science*, 359(6379): eaao0185.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Gu, K.; Shang, R.; Jiang, R.; Kuang, K.; Lin, R.-J.; Lyu, D.; Mao, Y.; Pan, Y.; Wu, T.; Yu, J.; et al. 2024. Blade: Benchmarking language model agents for data-driven science. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 13936–13971.

Gupta, J. S.; SI, H.; Singh, S. K.; Tawseeq, S. M.; Singla, Y. K.; Doermann, D.; Shah, R. R.; and Krishnamurthy, B. 2026. Accelerating Social Science Research via Agentic Hypothesization and Experimentation. *arXiv preprint arXiv:2602.07983*.

Huang, K.; Jin, Y.; Li, R.; Li, M. Y.; Candès, E.; and Leskovec, J. 2025. Automated hypothesis validation with agentic sequential falsifications. *arXiv preprint arXiv:2502.09858*.

K-Dense Inc. 2026. Astropy Skill for Scientific Agent Skills. Version 1.0, part of Scientific Agent Skills.

King, R. D.; Rowland, J.; Oliver, S. G.; Young, M.; Aubrey, W.; Byrne, E.; Liakata, M.; Markham, M.; Pir, P.; Soldatova, L. N.; et al. 2009. The automation of science. *Science*, 324(5923): 85–89.

Lakatos, I. 1978. *The methodology of scientific research programmes: Volume 1: Philosophical papers*, volume 1. Cambridge university press.

Langley, P. 1987. *Scientific discovery: Computational explorations of the creative processes*. MIT press.

Liu, H.; Huang, S.; Hu, J.; Zhou, Y.; and Tan, C. 2025. Hypobench: Towards systematic and principled benchmarking for hypothesis generation. *arXiv preprint arXiv:2504.11524*.

Liu, Y.; Su, Z.; Xie, L.; Zhang, Y.; Zong, Q.; Guo, J.; Xie, Z.; Ji, Y.; Yim, Y.; Luo, H.; et al. 2026. SkillRevise: Improving LLM-Authored Agent Skills via Trace-Conditioned Skill Revision. *arXiv preprint arXiv:2606.01139*.

Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; et al. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36: 46534–46594.

Majumder, B. P.; Surana, H.; Agarwal, D.; Dalvi Mishra, B.; Meena, A.; Prakhar, A.; Vora, T.; Khot, T.; Sabharwal, A.; and Clark, P. 2025. Discoverybench: Towards data-driven discovery with large language models. In *International Conference on Learning Representations*, volume 2025, 4556–4579.

Movva, R.; Peng, K.; Garg, N.; Kleinberg, J.; and Pierson, E. 2025. Sparse autoencoders for hypothesis generation. *arXiv preprint arXiv:2502.04382*.

Neuberg, L. G. 2003. Causality: models, reasoning, and inference, by judea pearl, cambridge university press, 2000. *Econometric Theory*, 19(4): 675–685.

Packer, C.; Fang, V.; Patil, S.; Lin, K.; Wooders, S.; and Gonzalez, J. 2023. MemGPT: towards LLMs as operating systems.

Park, J. S.; O'Brien, J.; Cai, C. J.; Morris, M. R.; Liang, P.; and Bernstein, M. S. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, 1–22.

Popper, K. 2005. *The logic of scientific discovery*. Routledge.

Shapere, D. 1964. The structure of scientific revolutions. *The Philosophical Review*, 73(3): 383–394.

Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.; and Yao, S. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36: 8634–8652.

Takagi, S.; Yamauchi, R.; and Kumagai, W. 2023. Towards autonomous hypothesis verification via language models with minimal guidance. *arXiv preprint arXiv:2311.09706*.

Vishe, Y.; Surana, R.; Jiang, X.; Huang, Z.; Li, X.; Kuang, N. L.; Yu, T.; Rossi, R. A.; Shang, J.; McAuley, J.; et al. 2026. Skill-r1: Agent skill evolution via reinforcement learning. *arXiv preprint arXiv:2605.09359*.

Wang, G.; Xie, Y.; Jiang, Y.; Mandlekar, A.; Xiao, C.; Zhu, Y.; Fan, L.; and Anandkumar, A. 2023a. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Wang, H.; Fu, T.; Du, Y.; Gao, W.; Huang, K.; Liu, Z.; Chandak, P.; Liu, S.; Van Katwyk, P.; Deac, A.; et al. 2023b. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972): 47–60.

Wang, J.; Yan, Q.; Wang, Y.; Tian, Y.; Mishra, S. S.; Xu, Z.; Gandhi, M.; Xu, P.; and Cheong, L. L. 2026. Reinforcement learning for self-improving agent with skill library. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1529–1550.

Yang, J.; Jimenez, C. E.; Wettig, A.; Lieret, K.; Yao, S.; Narasimhan, K. R.; and Press, O. 2024. Swe-agent: Agent-computer interfaces enable automated software engineering.

In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yang, Y.; Gong, Z.; Huang, W.; Yang, Q.; Zhou, Z.; Huang, Z.; Li, Y.; Gao, X.; Dai, Q.; Liu, B.; et al. 2026. Skillopt: Executive strategy for self-evolving agent skills. *arXiv preprint arXiv:2605.23904*.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Zhang, H.; Fan, S.; Zou, H. P.; Chen, Y.; Wang, Z.; Zhou, J.; Li, C.; Huang, W.-C.; Yao, Y.; Zheng, K.; et al. 2026. Coevoskills: Self-evolving agent skills via co-evolutionary verification. *arXiv preprint arXiv:2604.01687*.

Zhou, Y.; Liu, H.; Srivastava, T.; Mei, H.; and Tan, C. 2024. Hypothesis generation with large language models. In *Proceedings of the 1st Workshop on NLP for Science (NLP4Science)*, 117–139.