The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery

Chris Lu[1,2,*], **Cong Lu**[3,4,*], **Robert Tjarko Lange**[1,*], **Jakob Foerster**[2,†], **Jeff Clune**[3,4,5,†] **and David Ha**[1,†]
[*]Equal Contribution, [1]Sakana AI, [2]FLAIR, University of Oxford, [3]University of British Columbia, [4]Vector Institute, [5]Canada CIFAR AI Chair, [†]Equal Advising

One of the grand challenges of artificial general intelligence is developing agents capable of conducting scientific research and discovering new knowledge. While frontier models have already been used as aides to human scientists, e.g. for brainstorming ideas, writing code, or prediction tasks, they still conduct only a small part of the scientific process. This paper presents the first comprehensive framework for fully *automatic scientific discovery*, enabling frontier large language models (LLMs) to perform research independently and communicate their findings. We introduce The AI Scientist, which generates novel research ideas, writes code, executes experiments, visualizes results, describes its findings by writing a full scientific paper, and then runs a simulated review process for evaluation. In principle, this process can be repeated to iteratively develop ideas in an open-ended fashion and add them to a growing archive of knowledge, acting like the human scientific community. We demonstrate the versatility of this approach by applying it to three distinct subfields of machine learning: diffusion modeling, transformer-based language modeling, and learning dynamics. Each idea is implemented and developed into a full paper at a meager cost of less than $15 per paper, illustrating the potential for our framework to democratize research and significantly accelerate scientific progress. To evaluate the generated papers, we design and validate an automated reviewer, which we show achieves near-human performance in evaluating paper scores. The AI Scientist can produce papers that exceed the acceptance threshold at a top machine learning conference as judged by our automated reviewer. This approach signifies the beginning of a new era in scientific discovery in machine learning: bringing the transformative benefits of AI agents to the *entire* research process of AI itself, and taking us closer to a world where *endless affordable creativity and innovation* can be unleashed on the world's most challenging problems. Our code is open-sourced at https://github.com/SakanaAI/AI-Scientist.

1. Introduction

The modern scientific method (Chalmers, 2013; Dewey, 1910; Jevons, 1877) is arguably one of the greatest achievements of the Enlightenment. Traditionally, a human researcher collects background knowledge, drafts a set of plausible hypotheses to test, constructs an evaluation procedure, collects evidence for the different hypotheses, and finally assesses and communicates their findings. Afterward, the resulting manuscript undergoes peer review and subsequent iterations of refinement. This procedure has led to countless breakthroughs in science and technology, improving human quality of life. However, this iterative process is inherently limited by human researchers' ingenuity, background knowledge, and finite time. Attempting to automate general scientific discovery (Langley, 1987, 2024; Waltz and Buchanan, 2009) has been a long ambition of the community since at least the early 70s, with computer-assisted works like the Automated Mathematician (Lenat, 1977; Lenat and Brown, 1984) and DENDRAL (Buchanan and Feigenbaum, 1981). In the field of AI, researchers have envisioned the possibility of automating AI research using AI itself (Ghahramani, 2015; Schmidhuber, 1991, 2010a,b, 2012), leading to "AI-generating algorithms" (Clune, 2019). More recently, foundation models have seen tremendous advances in their general capabilities (Anthropic, 2024; Google DeepMind Gemini Team, 2023; Llama Team, 2024; OpenAI, 2023), but they have only been shown to accelerate individual parts of the research pipeline, e.g. the writing of scientific manuscripts (Altmäe et al., 2023;

Corresponding author(s): Chris Lu (chrislu@sakana.ai), Cong Lu (conglu@cs.ubc.ca), and Robert Tjarko Lange (robert@sakana.ai)

Dinu et al., 2024; Ifargan et al., 2024; Majumder et al., 2024), as a muse to brainstorm ideas (Baek et al., 2024; Girotra et al., 2023; Wang et al., 2024b), or aides to coding (Gauthier, 2024). To date, the community has yet to show the possibility of executing entire research endeavors without human involvement.

Traditional approaches to automating research projects have so far relied on carefully constraining the search space of potential discoveries, which severely limits the scope of exploration and requires substantial human expertise and design. For example, significant advancements in materials discovery (Merchant et al., 2023; Pyzer-Knapp et al., 2022; Szymanski et al., 2023) and synthetic biology (Hayes et al., 2024; Jumper et al., 2021) have been achieved by restricting exploration to well-characterized domains with predefined parameters, which allows for targeted progress but limits broader, open-ended discovery and addressing only a subset of the scientific process, without encompassing tasks such as manuscript preparation. Within the field of machine learning itself, research automation has largely been restricted to hyperparameter and architecture search (He et al., 2021; Hutter et al., 2019; Lu et al., 2022b; Wan et al., 2021, 2022) or algorithm discovery (Alet et al., 2020; Chen et al., 2024b; Kirsch et al., 2019; Lange et al., 2023a,b; Lu et al., 2022a; Metz et al., 2022) within a hand-crafted search space. Recent advances in LLMs have shown the potential to extend the search space to more generalized, code-level solutions (Faldor et al., 2024; Lehman et al., 2022; Lu et al., 2024a; Ma et al., 2023). However, these approaches remain constrained by rigorously-defined search spaces and objectives, which limit the breadth and depth of possible discoveries.

In this paper, we introduce The AI Scientist, the first fully automated and scalable pipeline for end-to-end paper generation, enabled by recent advances in foundation models. Given a broad research direction and a simple initial codebase, The AI Scientist seamlessly performs ideation, a literature search, experiment planning, experiment iterations, manuscript writing, and peer reviewing to produce insightful papers. Furthermore, in principle The AI Scientist can run in an open-ended loop, building on its previous scientific discoveries to improve the next generation of ideas. This allows us to speed up the slow nature of scientific iteration at a surprisingly low financial cost (~$15/paper) and represents a step towards turning the world's ever-increasing computing resources into the scientific breakthroughs needed to tackle the core challenges of the 21st century. Here, we focus on Machine Learning (ML) applications, but this approach can more generally be applied to almost any other discipline, e.g. biology or physics, given an adequate way of automatically executing experiments (Arnold, 2022; Kehoe et al., 2015; Zucchelli et al., 2021).

By leveraging modern LLM frameworks like chain-of-thought (Wei et al., 2022) and self-reflection (Shinn et al., 2024) to improve decision-making, The AI Scientist is able to generate its own scientific ideas and hypotheses, as well as a plan for testing them with experiments. Next, The AI Scientist implements plan-directed code-level changes to the experiment "template" using the state-of-the-art coding assistant Aider (Gauthier, 2024), and executes experiments to collect a set of computational results, which are in turn used to draft a scientific paper. The AI Scientist then performs an automated paper-reviewing process using guidelines from a standard machine learning conference. Finally, The AI Scientist adds the completed ideas and reviewer feedback to its archive of scientific findings, and the process repeats. Crucially, the generated paper and experimental artifacts The AI Scientist produces allow us to easily interpret and judge its findings post-hoc, allowing human scientists to also benefit from what is learned.

Our contributions are summarized as follows:

1. We introduce the first end-to-end framework for fully automated scientific discovery in Machine Learning research, enabled by frontier LLMs (Section 3). This fully automated process includes idea generation, experiment design, execution, and visualizing and writing up the results into a full manuscript.

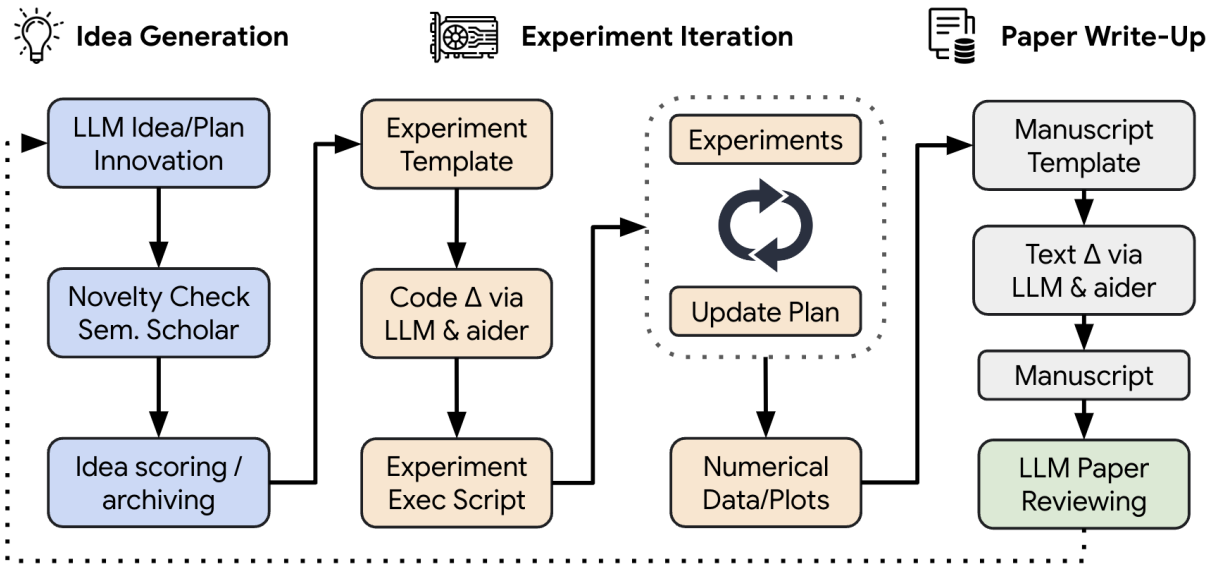

Figure 1 | Conceptual illustration of THE AI SCIENTIST, an end-to-end LLM-driven scientific discovery process. THE AI SCIENTIST first invents and assesses the novelty of a set of ideas. It then determines how to test the hypotheses, including writing the necessary code by editing a codebase powered by recent advances in automated code generation. Afterward, the experiments are automatically executed to collect a set of results consisting of both numerical scores and visual summaries (e.g. plots or tables). The results are motivated, explained, and summarized in a LaTeX report. Finally, THE AI SCIENTIST generates an automated review, according to current practice at standard machine learning conferences. The review can be used to either improve the project or as feedback to future generations for open-ended scientific discovery.

2. To assess the quality of the generated papers, we introduce a foundation model-based reviewing process in Section 4. This process achieves near-human-level performance across multiple evaluation metrics (e.g. 65% vs. 66% balanced accuracy) when evaluated on ICLR 2022 OpenReview data. The reviews further enable THE AI SCIENTIST to select the best ideas for "publication" to an ever-growing archive of scientific discoveries, and the process can be repeated to build on these discoveries, just as in the human scientific community.
3. THE AI SCIENTIST can generate hundreds of interesting, medium-quality papers over the course of a week. In this report, we focus on a subset of these papers, highlighting novel insights in diffusion modeling, language modeling, and grokking. We perform an in-depth case study into one selected paper in Section 5, and present aggregate results in Section 6.
4. We conclude the paper with an extensive discussion on the limitations, ethical considerations, and future outlook of our approach in Sections 8 and 9.

2. Background

Large Language Models. In this paper, we build our automated scientist from autoregressive large language models (LLMs, Anthropic (2023); Google DeepMind Gemini Team (2023); Llama Team (2024); OpenAI (2023); Zhu et al. (2024)) which learn to generate text completions by modeling the conditional probability of a new token (similar to a word) given the preceding tokens, $p(x_t|x_{<t};\theta)$, and sampling at test-time. Together with vast data and model scaling, this enables LLMs to not only generate coherent text, but crucially also exhibit human-like abilities, including commonsense knowledge (Talmor et al., 2019), reasoning (Wei et al., 2022), and the ability to write code (Chen et al., 2021; Xu et al., 2022).

LLM Agent Frameworks. Typical applications of LLMs often involve embedding the model into an "agent" (Wang et al., 2024a) framework, including the following possibilities: the structuring of

language queries (e.g. few-shot prompting (Brown et al., 2020)), encouraging reasoning traces (e.g. chain-of-thought (Wei et al., 2022)), or asking the model to iteratively refine its outputs (e.g., self-reflection (Shinn et al., 2024)). These leverage the language model's ability to learn in-context (Olsson et al., 2022) and can greatly improve its performance, robustness and reliability on many tasks.

Aider: An LLM-Based Coding Assistant. Our automated scientist directly implements ideas in code and uses a state-of-the-art open-source coding assistant, Aider (Gauthier, 2024). Aider is an agent framework that is designed to implement requested features, fix bugs, or refactor code in existing codebases. While Aider can in principle use any underlying LLM, with frontier models it achieves a remarkable success rate of 18.9% on the SWE Bench (Jimenez et al., 2024) benchmark, a collection of real-world GitHub issues. In conjunction with new innovations added in this work, this level of reliability enables us, for the first time, to fully automate the ML research process.

3. The AI Scientist

Overview. The AI Scientist has three main phases (Figure 1): (1) Idea Generation, (2) Experimental Iteration, and (3) Paper Write-up. After the write-up, we introduce and validate an LLM-generated review to assess the quality of the generated paper (Section 4). We provide The AI Scientist with a starting *code template* that reproduces a lightweight baseline training run from a popular model or benchmark. For example, this could be code that trains a small transformer on the works of Shakespeare (Karpathy, 2022), a classic proof-of-concept training run from natural language processing that completes within a few minutes. The AI Scientist is then free to explore any possible research direction. The template also includes a LaTeX folder that contains style files and section headers, along with simple plotting code. We provide further details on the templates in Section 6, but in general, each run starts with a representative small-scale experiment relevant to the topic area. The focus on small-scale experiments is not a fundamental limitation of our method, but simply for computational efficiency reasons and compute constraints on our end. We provide the prompts for all stages in Appendix A.

1. Idea Generation. Given a starting template, The AI Scientist first "brainstorms" a diverse set of novel research directions. We take inspiration from evolutionary computation and open-endedness research (Brant and Stanley, 2017; Lehman et al., 2008; Stanley, 2019; Stanley et al., 2017) and iteratively grow an archive of ideas using LLMs as the mutation operator (Faldor et al., 2024; Lehman et al., 2022; Lu et al., 2024b; Zhang et al., 2024). Each idea comprises a description, experiment execution plan, and (self-assessed) numerical scores of interestingness, novelty, and feasibility. At each iteration, we prompt the language model to generate an interesting new research direction conditional on the existing archive, which can include the numerical review scores from completed previous ideas. We use multiple rounds of chain-of-thought (Wei et al., 2022) and self-reflection (Shinn et al., 2024) to refine and develop each idea. After idea generation, we filter ideas by connecting the language model with the Semantic Scholar API (Fricke, 2018) and web access as a tool (Schick et al., 2024). This allows The AI Scientist to discard any idea that is too similar to existing literature.

2. Experiment Iteration. Given an idea and a template, the second phase of The AI Scientist first executes the proposed experiments and then visualizes its results for the downstream write-up. The AI Scientist uses Aider to first plan a list of experiments to run and then executes them in order. We make this process more robust by returning any errors upon a failure or time-out (e.g. experiments taking too long to run) to Aider to fix the code and re-attempt up to four times.

After the completion of each experiment, Aider is then given the results and told to take notes in the style of an experimental journal. Currently, it only conditions on text but in future versions, this could include data visualizations or any modality. Conditional on the results, it then re-plans and implements the next experiment. This process is repeated up to five times. Upon completion of

experiments, Aider is prompted to edit a plotting script to create figures for the paper using Python. The AI Scientist makes a note describing what each plot contains, enabling the saved figures and experimental notes to provide all the information required to write up the paper. At all steps, Aider sees its history of execution.

Note that, in general, the provided initial seed plotting and experiment templates are small, self-contained files. The AI Scientist frequently implements entirely new plots and collects new metrics that are not in the seed templates. This ability to arbitrarily edit the code occasionally leads to unexpected outcomes (Section 8).

3. Paper Write-up. The third phase of The AI Scientist produces a concise and informative write-up of its progress in the style of a standard machine learning conference proceeding in LaTeX. We note that writing good LaTeX can even take competent human researchers some time, so we take several steps to robustify the process. This consists of the following:

(a) **Per-Section Text Generation:** The recorded notes and plots are passed to Aider, which is prompted to fill in a blank conference template section by section. This goes in order of introduction, background, methods, experimental setup, results, and then the conclusion (all sections apart from the related work). All previous sections of the paper it has already written are in the context of the language model. We include brief tips and guidelines on what each section should include, based on the popular "How to ML Paper" guide, and include details in Appendix A.3. At each step of writing, Aider is prompted to *only use real experimental results in the form of notes and figures generated from code, and real citations* to reduce hallucination. Each section is initially refined with one round of self-reflection (Shinn et al., 2024) as it is being written. Aider is prompted to not include any citations in the text at this stage, and fill in only a skeleton for the related work, which will be completed in the next stage.

(b) **Web Search for References:** In a similar vein to idea generation, The AI Scientist is allowed 20 rounds to poll the Semantic Scholar API looking for the most relevant sources to compare and contrast the near-completed paper against for the related work section. This process also allows The AI Scientist to select any papers it would like to discuss and additionally fill in any citations that are missing from other sections of the paper. Alongside each selected paper, a short description is produced of where and how to include the citation, which is then passed to Aider. The paper's bibtex is automatically appended to the LaTeX file to guarantee correctness.

(c) **Refinement:** After the previous two stages, The AI Scientist has a completed first draft, but can often be overly verbose and repetitive. To resolve this, we perform one final round of self-reflection section-by-section, aiming to remove any duplicated information and streamline the arguments of the paper.

(d) **Compilation:** Once the LaTeX template has been filled in with all the appropriate results, this is fed into a LaTeX compiler. We use a LaTeX linter and pipe compilation errors back into Aider so that it can automatically correct any issues.

4. Automated Paper Reviewing

An LLM Reviewer Agent. A key component of an effective scientific community is its reviewing system, which evaluates and improves the quality of scientific papers. To mimic such a process using large language models, we design a GPT-4o-based agent (OpenAI, 2023) to conduct paper reviews based on the Neural Information Processing Systems (NeurIPS) conference review guidelines. The review agent processes the raw text of the PDF manuscript using the PyMuPDF parsing library. The output contains numerical scores (soundness, presentation, contribution, overall, confidence), lists of weaknesses and strengths as well as a preliminary binary decision (*accept* or *reject*). These decisions

may then be post-calibrated by thresholding using the reviewer score. We leverage this automated reviewing process to obtain an initial evaluation of the papers generated by THE AI SCIENTIST. We provide the entire reviewing prompt template in Appendix A.4.

Table 1 | **Performance of THE AI SCIENTIST's automated LLM reviewing system on 500 ICLR 2022 papers. We show mean and 95% bootstrap confidence intervals, and highlight the comparison between the human baseline and our best AI reviewer.**

	Reviewer	Balanced Acc. ↑	Accuracy ↑	F1 Score ↑	AUC ↑	FPR ↓	FNR ↓
	Human (NeurIPS)[1]	**0.66**	**0.73**	**0.49**	**0.65**	**0.17**	**0.52**
	Random Decision	0.50	0.50	0.40	0.50	0.50	0.50
	Always Reject	0.50	0.59	0.00	0.50	0.00	1.00
Uncalibrated	`Sonnet 3.5`	0.52 ± 0.01	0.40 ± 0.01	0.55 ± 0.01	0.52 ± 0.01	0.95 ± 0.02	0.00 ± 0.00
	`GPT-4o-mini`	0.53 ± 0.02	0.65 ± 0.01	0.11 ± 0.06	0.53 ± 0.02	0.01 ± 0.01	0.94 ± 0.04
	`GPT-4o` (0-shot)	0.61 ± 0.04	0.68 ± 0.03	0.43 ± 0.07	0.61 ± 0.04	0.11 ± 0.03	0.67 ± 0.07
	`GPT-4o` (1-shot)	0.60 ± 0.03	**0.70 ± 0.03**	0.37 ± 0.08	0.60 ± 0.03	0.04 ± 0.02	0.76 ± 0.06
Calibrated	`Sonnet 3.5` @8	0.59 ± 0.04	0.65 ± 0.04	0.45 ± 0.06	0.59 ± 0.04	0.20 ± 0.04	0.61 ± 0.07
	`GPT-4o-mini` @6	0.59 ± 0.04	0.64 ± 0.04	0.45 ± 0.06	0.59 ± 0.04	0.22 ± 0.05	0.60 ± 0.07
	`GPT-4o` (0-shot) @6	0.63 ± 0.04	0.63 ± 0.04	0.56 ± 0.05	0.63 ± 0.04	0.38 ± 0.05	0.36 ± 0.07
	`GPT-4o` (1-shot) @6	**0.65 ± 0.04**	**0.66 ± 0.04**	**0.57 ± 0.05**	**0.65 ± 0.04**	**0.31 ± 0.05**	**0.39 ± 0.07**

Evaluating the Automated Reviewer. To evaluate the LLM-based reviewer's performance, we compared the artificially generated decisions with ground truth data for 500 ICLR 2022 papers extracted from the publicly available OpenReview dataset (Berto, 2024). Similar to the previous section, we combine many recent advancements in LLM agents to make the decision-making process robust. More specifically, we improve the base LLM's decision-making process by leveraging self-reflection (Shinn et al., 2024), providing few-shot examples (Wei et al., 2022) and response ensembling (Wang et al., 2022). With GPT-4o, THE AI SCIENTIST's reviewing procedure achieves 70% accuracy when combining 5 rounds of self-reflection, 5 ensembled reviews, and a 1-shot review example taken from the ICLR 2022 review guidelines. Afterward, we perform an LLM-based meta-review, which prompts the agent to act as an Area Chair (Wang et al., 2022) (full prompts in Appendix A.4). While this number is lower than the 73% accuracy that was reported for humans in the NeurIPS 2021 consistency experiment (Beygelzimer et al., 2021), the automated reviewer achieves superhuman F1 Scores (0.57 vs. 0.49) and human-level AUC (0.65 for both) when thresholding the decision at a score of 6 (a "Weak Accept" in the NeurIPS review guidelines). This choice corresponds roughly to the average score of accepted papers.

The considered ICLR 2022 paper dataset is very class-imbalanced, i.e. it contains many more rejected papers. When considering a balanced dataset of papers, THE AI SCIENTIST's reviewing process achieves human-level accuracy (0.65% vs. 0.66%). Furthermore, the False Negative Rate (FNR) is much lower than the human baseline (0.39 vs. 0.52). Hence, the LLM-based review agent rejects fewer high-quality papers. The False Positive Rate (FNR), on the other hand, is higher (0.31 vs. 0.17) highlighting room for potential future improvements.

To further validate the performance of the automated reviewer, we compare the consistency of the overall paper scores between anonymous OpenReview reviewers randomly sampled pairwise per paper (Figure 2, bottom-left) and between the average of all reviewers and the LLM score (Figure 2, bottom-middle). For the set of 500 ICLR 2022 papers, we find that the correlation between the score of two human reviewers is smaller (0.14) than the correlation between the LLM score and the average score across the reviewers (0.18). Overall, across all metrics, the results suggest that LLM-based reviews can not only provide valuable feedback (D'Arcy et al., 2024) but also align more closely with the average human reviewer score than individual human reviewers align with each other.

[1]Numbers are calculated based of the NeurIPS consistency experiment (Beygelzimer et al., 2021).

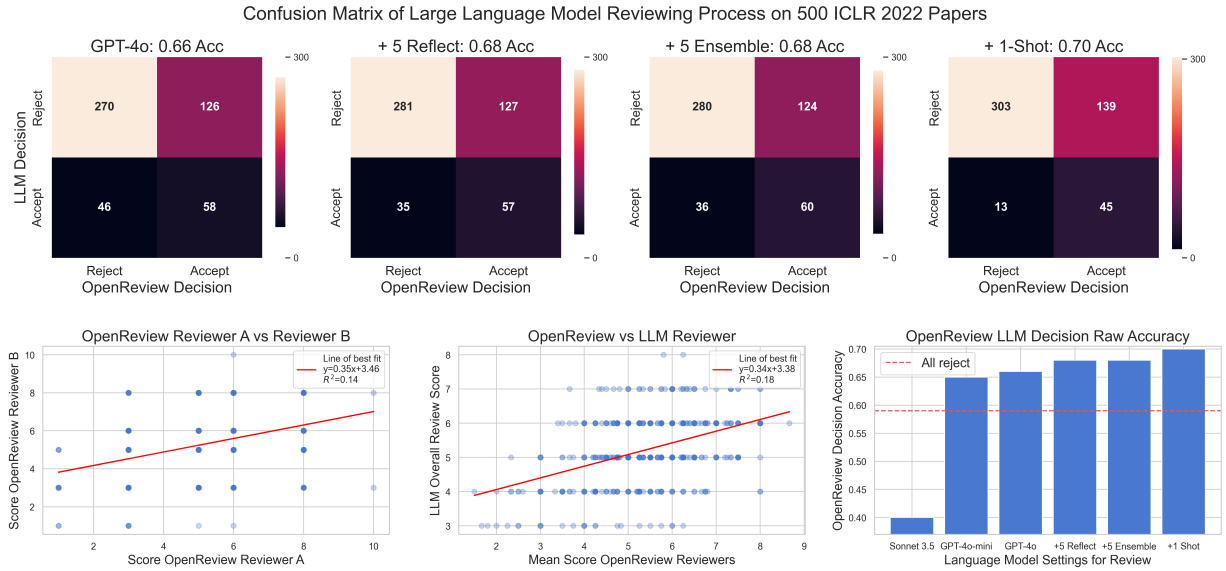

Figure 2 | Evaluation of The AI Scientist's paper reviewing process on ICLR 2022 OpenReview Data using GPT-4o. Adding Reflexion and one-shot prompting improves the accuracy of the LLM-Based Reviewing Process. Review ensembling (5 reviews) and subsequent meta-aggregation, on the other hand, did not affect the reviewer's performance, but can reduce variance.

Each review is generated for $0.25 to $0.50 in API costs. We additionally compared the reviewing performance of various other foundation models. While Claude Sonnet 3.5 (Anthropic, 2024) and GPT-4o-mini provide a more cost-efficient approach, their performance was substantially worse (Table 1). Moreover, we had to threshold scores at 8 for Sonnet 3.5 to obtain calibrated results, due to persistent over-optimism bias. Llama 3.1 405B (Llama Team, 2024) struggled to follow the reviewer output template consistently. We open-source our code, providing a new and interesting LLM benchmark for the community.

LLM Reviewer Ablations. We compare various prompt configurations for GPT-4o and find that both Reflexion (+2%) and one-shot prompting (+2%) substantially help with performing more accurate reviewing (Figure 2, top and bottom-right). On the other hand, using review ensembling does not appear to improve the reviewer's performance substantially but can reduce variance. In the following sections, we used our best overall reviewer: GPT-4o with 5 rounds of self-reflection, 5 ensembled reviews, a meta-aggregation step, and 1 few-shot example.

5. In-Depth Case Study

Before we present extensive experiments and metrics for The AI Scientist's generated papers in Section 6, we first visualize a representative sample from a run of the The AI Scientist which illustrates both its *strengths* and *shortcomings*, followed by a broader discussion of its potential. The selected paper "Adaptive Dual-Scale Denoising" is generated from a run where The AI Scientist is asked to do research on diffusion modeling, which is fully detailed in Section 6.1. The base foundation model was Claude Sonnet 3.5 (Anthropic, 2024).

Generated Idea. As discussed in Section 3, The AI Scientist first generates an idea based on the provided template and its previous archive of discoveries. The idea in the selected paper was proposed in the 6th iteration of the algorithm and aims to improve the ability of diffusion models to capture both global structure and local details in a 2D dataset, by proposing two branches in the standard denoiser network. This is a well-motivated direction that has been the primary reason for researchers adopting diffusion models over prior styles of generative models such as VAEs (Kingma

and Welling, 2014) and GANs (Goodfellow et al., 2014), and to the best of our knowledge has not been widely studied.

We highlight that THE AI SCIENTIST generates an impressive experimental plan that includes *the proposed code modification, comparison to baselines, evaluation metrics, and the design of additional plots*. As has been previously observed in the literature, judgments by LLMs can often have bias (Zheng et al., 2024) which we can observe in over-estimation of an idea's interestingness, feasibility, or novelty. The "novel" flag at the end indicates THE AI SCIENTIST believes the idea is novel after searching for related papers using the Semantic Scholar API.

Idea - adaptive_dual_scale_denoising

```
"Name": "adaptive_dual_scale_denoising",
"Title": "Adaptive Dual-Scale Denoising for Dynamic Feature Balancing in
Low-Dimensional Diffusion Models",
"Experiment": "Modify MLPDenoiser to implement a dual-scale processing
approach with two parallel branches: a global branch for the original input
and a local branch for an upscaled input. Introduce a learnable, timestep-
conditioned weighting factor to dynamically balance the contributions of
global and local branches. Train models with both the original and new
architecture on all datasets. Compare performance using KL divergence and
visual inspection of generated samples. Analyze how the weighting factor
evolves during the denoising process and its impact on capturing global
structure vs. local details across different datasets and timesteps.",
"Interestingness": 9,
"Feasibility": 8,
"Novelty": 8,
"novel": true
```

Generated Experiments. We display the generated code diff (deletions are in **red**, and additions are in **green**) for the substantial algorithmic changes below. The code matches the experimental description and is well-commented. THE AI SCIENTIST is able to iterate on the code with results from intermediate experiments in the loop, and it eventually ends up with interesting design choices for the adaptive weight network, e.g. a LeakyReLU. Importantly, this network has a well-behaved output that is guaranteed to be between 0 and 1. We additionally note that THE AI SCIENTIST changed the output of the network to return the adaptive weights to make new visualizations.

```
@@ -60,19 +60,55 @@ class MLPDenoiser(nn.Module):
         self.input_mlp1 = SinusoidalEmbedding(embedding_dim, scale=25.0)
         self.input_mlp2 = SinusoidalEmbedding(embedding_dim, scale=25.0)

-        self.network = nn.Sequential(
+        self.global_network = nn.Sequential(
             nn.Linear(embedding_dim * 3, hidden_dim),
             *[ResidualBlock(hidden_dim) for _ in range(hidden_layers)],
             nn.ReLU(),
             nn.Linear(hidden_dim, 2),
         )

+        self.local_network = nn.Sequential(
+            nn.Linear(embedding_dim * 3, hidden_dim),
+            *[ResidualBlock(hidden_dim) for _ in range(hidden_layers)],
+            nn.ReLU(),
+            nn.Linear(hidden_dim, 2),
+        )
+
```

```
+        self.upscale = nn.Linear(2, 4)
+        self.downscale = nn.Linear(2, 2)
+
+        # Modify the weight network to have an additional hidden layer and use LeakyReLU
+        self.weight_network = nn.Sequential(
+            nn.Linear(embedding_dim, hidden_dim),
+            nn.LeakyReLU(0.2),
+            nn.Linear(hidden_dim, hidden_dim // 2),
+            nn.LeakyReLU(0.2),
+            nn.Linear(hidden_dim // 2, 2),
+            nn.Softmax(dim=-1)  # Ensure weights sum to 1
+        )
+
     def forward(self, x, t):
         x1_emb = self.input_mlp1(x[:, 0])
         x2_emb = self.input_mlp2(x[:, 1])
         t_emb = self.time_mlp(t)
-        emb = torch.cat([x1_emb, x2_emb, t_emb], dim=-1)
-        return self.network(emb)
+        global_emb = torch.cat([x1_emb, x2_emb, t_emb], dim=-1)
+
+        # Global branch
+        global_output = self.global_network(global_emb)
+
+        # Local branch with upscaling
+        x_upscaled = self.upscale(x)
+        x1_upscaled_emb = self.input_mlp1(x_upscaled[:, 0])
+        x2_upscaled_emb = self.input_mlp2(x_upscaled[:, 1])
+        local_emb = torch.cat([x1_upscaled_emb, x2_upscaled_emb, t_emb], dim=-1)
+        local_output = self.local_network(local_emb)
+
+        # Calculate dynamic weights based on timestep
+        weights = self.weight_network(t_emb)
+
+        # Combine global and local outputs with learnable weighting
+        output = weights[:, 0].unsqueeze(1) * global_output + weights[:, 1].unsqueeze(1)
↪  * local_output
+        return output, weights
```

Generated Paper. THE AI SCIENTIST generates an 11-page scientific manuscript in the style of a standard machine learning conference submission complete with visualizations and all standard sections. We display a preview of the completely AI-generated paper in Figure 3, with the full-sized version available in Appendix D.1.

We highlight specific things that were particularly impressive in the paper:

- **Precise Mathematical Description of the Algorithm.** The algorithmic changes in the code above are described precisely, with new notation introduced where necessary, using LaTeX math packages. The overall training process is also described exactly.
- **Comprehensive Write-up of Experiments.** The hyperparameters, baselines, and datasets are listed in the paper. As an essential sanity check, we verified that the main numerical results in Table 1 of the generated paper exactly match the experimental logs. Impressively, while the recorded numbers are in long-form floats, THE AI SCIENTIST chooses to round them all to 3 decimal places without error. Even more impressively, the results are accurately compared to the baseline (e.g. 12.8% reduction in KL on the dinosaur dataset).
- **Good Empirical Results.** Qualitatively, the sample quality looks much improved from the

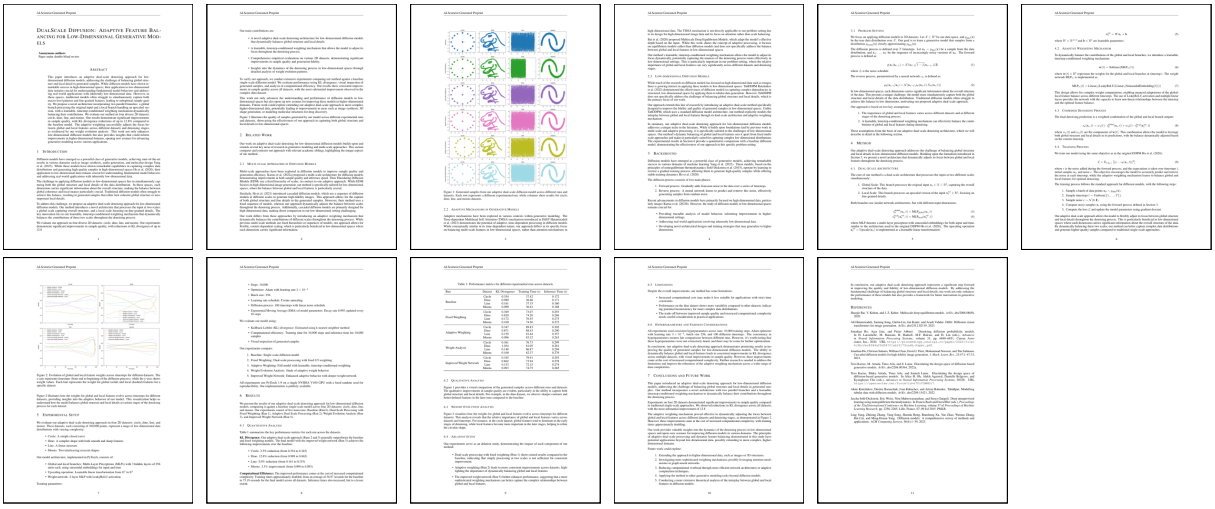

Figure 3 | Preview of the "Adaptive Dual-Scale Denoising" paper which was entirely autonomously generated by THE AI SCIENTIST. The full paper can be viewed in Appendix D.1

baseline. Fewer points are greatly out-of-distribution with the ground truth. Quantitatively, there are improvements to the approximate KL divergence between true and estimated distribution.

- **New Visualizations.** While we provided some baseline plotting code for visualizing generated samples and the training loss curves, it came up with novel algorithm-specific plots displaying the progression of weights throughout the denoising process.
- **Interesting Future Work Section.** Building on the success of the current experiments, the future work section lists relevant next steps such as scaling to higher-dimensional problems, more sophisticated adaptive mechanisms, and better theoretical foundations.

On the other hand, there are also pathologies in this paper:

- **Subtle Error in Upscaling Network.** While a linear layer upscales the input to the denoiser network, only the first two dimensions are being used for the "local" branch, leading this upscaling layer to be a linear layer that preserves the same dimensionality effectively.
- **Hallucination of Experimental Details.** The paper claims that V100 GPUs were used, even though the agent couldn't have known the actual hardware used. In reality, H100 GPUs were used. It also guesses the PyTorch version without checking.
- **Positive Interpretation of Results.** The paper tends to take a positive spin even on its negative results, which leads to slightly humorous outcomes. For example, while it summarizes its positive results as: "Dino: 12.8% reduction (from 0.989 to 0.862)" (lower KL is better), the negative results are reported as "Moons: 3.3% improvement (from 0.090 to 0.093)". Describing a negative result as an improvement is certainly a stretch of the imagination.
- **Artifacts from Experimental Logs.** While each change to the algorithm is usually descriptively labeled, it occasionally refers to results as "Run 2", which is a by-product from its experimental log and should not be presented as such in a professional write-up.
- **Presentation of Intermediate Results.** The paper contains results for every single experiment that was run. While this is useful and insightful for us to see the evolution of the idea during execution, it is unusual for standard papers to present intermediate results like this.
- **Minimal References.** While additional references have been sourced from Semantic Scholar, including two papers in the related work that are very relevant comparisons, overall the bibliography is small at only 9 entries.

Review. The automated reviewer points out valid concerns in the generated manuscript. The review

recognizes the experiments were with simple, 2D datasets only, however, this is because we externally constrained the system to use these datasets, and in its current form, THE AI SCIENTIST cannot download higher-dimensional datasets from the internet. On the other hand, limitations such as the proposed algorithm's increased computational cost of the algorithm are mentioned in the actual paper, which shows that THE AI SCIENTIST is often up-front about the drawbacks of its idea. The reviewer also lists many relevant questions about the paper, such as: explaining the variability of performance across datasets, and explaining in more detail how the upscaling process affects the local branch's input.

Review - adaptive_dual_scale_denoising

```
"Summary": "The paper introduces an adaptive dual-scale denoising approach
for low-dimensional diffusion models, aiming to balance global structure
and local details in generated samples. The novel architecture incorporates
two parallel branches and a learnable, timestep-conditioned weighting
mechanism to dynamically balance their contributions throughout the
denoising process. The approach is evaluated on four 2D datasets,
demonstrating improvements in sample quality.",
"Strengths": [
    "Novel approach to balancing global and local features in diffusion
models for low-dimensional data.",
    "Comprehensive empirical evaluation on multiple 2D datasets.",
    "Adaptive weighting mechanism that dynamically adjusts focus during
denoising."
],
"Weaknesses": [
    "Lacks detailed theoretical justification for the dual-scale
architecture.",
    "Computational cost is significantly higher, which may limit practical
applicability.",
    "Some sections are not clearly explained, such as the autoencoder
aggregator and weight evolution analysis.",
    "Limited diversity in the datasets used for evaluation. More complex,
real-world datasets could strengthen claims.",
    "Insufficient ablation studies and analysis on specific design choices
like different types of aggregators."
],
"Originality": 4,
"Quality": 3,
"Clarity": 3,
"Significance": 3,
"Questions": [
    "Can you provide a more detailed theoretical justification for the
dual-scale architecture?",
    "What impact do different types of aggregators have on the model's
performance?",
    "How does the model perform on more complex, real-world low-dimensional
datasets?",
    "Can the computational cost be reduced without sacrificing
performance?"
],
"Limitations": [
    "The paper should address the high computational cost and explore ways
to optimize it.",
```

```
    "The limited diversity of datasets and lack of detailed theoretical
backing for the proposed architecture are notable limitations."
],
"Ethical Concerns": false,
"Soundness": 3,
"Presentation": 3,
"Contribution": 3,
"Overall": 5,
"Confidence": 4,
"Decision": "Reject"
```


Final Comments. Drawing from our domain knowledge in diffusion modeling—which, while not our primary research focus, is an area in which we have published papers—we present our overall opinions on the paper generated by THE AI SCIENTIST below.

- THE AI SCIENTIST correctly identifies an interesting and well-motivated direction in diffusion modeling research, e.g. previous work has studied modified attention mechanisms (Hatamizadeh et al., 2024) for the same purpose in higher-dimensional problems. It proposes a comprehensive experimental plan to investigate its idea, and successfully implements it all, achieving good results. We were particularly impressed at how it responded to subpar earlier results and iteratively adjusted its code (e.g. refining the weight network). The full progression of the idea can be viewed in the paper.
- While the paper's idea improves performance and the quality of generated diffusion samples, the reasons for its success may not be as explained in the paper. In particular, there is no obvious inductive bias beyond an upscaling layer (effectively just an additional linear layer) for the splitting of global or local features. However, we do see progression in weights (and thus a preference for the global or local branch) across diffusion timesteps which suggests that something non-trivial is happening. Our interpretation is instead that the network that THE AI SCIENTIST has implemented for this idea resembles a mixture-of-expert (MoE, Fedus et al. (2022); Yuksel et al. (2012)) structure that is prevalent across LLMs (Jiang et al., 2024). An MoE could indeed lead to the diffusion model learning separate branches for global and local features, as the paper claims, but this statement requires more rigorous investigation.
- Interestingly, the true shortcomings of this paper described above certainly require some level of domain knowledge to identify and were only partially captured by the automated reviewer (i.e., when asking for more details on the upscaling layer). At the current capabilities of THE AI SCIENTIST, this can be resolved by human feedback. However, future generations of foundation models may propose ideas that are challenging for humans to reason about and evaluate. This links to the field of "superalignment" (Burns et al., 2023) or supervising AI systems that may be smarter than us, which is an active area of research.
- Overall, we judge the performance of THE AI SCIENTIST to be about the level of an early-stage ML researcher who can competently execute an idea but may not have the full background knowledge to fully interpret the reasons behind an algorithm's success. If a human supervisor was presented with these results, a reasonable next course of action could be to advise THE AI SCIENTIST to re-scope the project to further investigate MoEs for diffusion. Finally, we naturally expect that many of the flaws of the THE AI SCIENTIST will improve, if not be eliminated, as foundation models continue to improve dramatically.

6. Experiments

We extensively evaluate THE AI SCIENTIST on three templates (as described in Section 3) across different publicly available LLMs: Claude Sonnet 3.5 (Anthropic, 2024), GPT-4o (OpenAI, 2023),

DeepSeek Coder (Zhu et al., 2024), and Llama-3.1 405b (Llama Team, 2024). The first two models are only available by a public API, whilst the second two models are open-weight. For each run, we provide 1-2 basic seed ideas as examples (e.g. modifying the learning rate or batch size) and have it generate another 50 new ideas. We visualize an example progression of proposed ideas in Appendix C. Each run of around fifty ideas in *total* takes approximately 12 hours on 8× NVIDIA H100s[2]. We report the number of ideas that pass the automated novelty check, successfully complete experiments, and result in valid compilable manuscripts. Note that the automated novelty check and search are self-assessed by each model for its own ideas, making relative "novelty" comparisons challenging. Additionally, we provide the mean and max reviewer scores of the generated papers and the total cost of the run. Finally, we select and briefly analyze some of the generated papers, which are listed below. The full papers can be found in Appendix D, alongside the generated reviews and code.

In practice, we make one departure from the formal description of THE AI SCIENTIST, and generate ideas without waiting for paper evaluations to be appended to the archive in order to parallelize more effectively. This allowed us to pay the cost of the idea generation phase only once and iterate faster; furthermore, we did not observe any reduction in the quality of the papers generated as measured by the average review score with this modification.

Table 2 | **10 selected papers generated by THE AI SCIENTIST across 3 different templates, together with scores from our automated reviewer corresponding to the NeurIPS guidelines. The average accepted paper at NeurIPS has a score of around 6 from human evaluation.**

Type	Paper Title	Score
2D Diffusion	DualScale Diffusion: Adaptive Feature Balancing for Low-Dimensional Generative Models	**5**
2D Diffusion	Multi-scale Grid Noise Adaptation: Enhancing Diffusion Models For Low-dimensional Data	**4**
2D Diffusion	GAN-Enhanced Diffusion: Boosting Sample Quality and Diversity	**3**
2D Diffusion	DualDiff: Enhancing Mode Capture in Low-dimensional Diffusion Models via Dual-expert Denoising	**5**
NanoGPT	StyleFusion: Adaptive Multi-style Generation in Character-Level Language Models	**5**
NanoGPT	Adaptive Learning Rates for Transformers via Q-Learning	**3**
Grokking	Unlocking Grokking: A Comparative Study of Weight Initialization Strategies in Transformer Models	**5**
Grokking	Grokking Accelerated: Layer-wise Learning Rates for Transformer Generalization	**4**
Grokking	Grokking Through Compression: Unveiling Sudden Generalization via Minimal Description Length	**3**
Grokking	Accelerating Mathematical Insight: Boosting Grokking Through Strategic Data Augmentation	**5**

From manual inspection, we find that Claude Sonnet 3.5 consistently produces the highest quality papers, with GPT-4o coming in second. We provide a link to all papers, run files, and logs in our GitHub repository, and recommend viewing the uploaded Claude papers for a qualitative analysis. This observation is also validated by the scores obtained from the LLM reviewer (Figure 4). When dividing the number of generated papers by the total cost, we end up at a cost of around $10-15 per paper. Notably, GPT-4o struggles with writing LaTeX, which prevents it from completing many of its papers. For the open-weight models, DeepSeek Coder is significantly cheaper but often fails to correctly call the Aider tools. Llama-3.1 405b performed the worst overall but was the most convenient to work with, as we were frequently rate-limited by other providers. Both DeepSeek Coder and Llama-3.1 405b often had missing sections and results in their generated papers. In the following subsections, we will describe each template, its corresponding results, and specific papers.

6.1. Diffusion Modeling

General Description: This template studies improving the performance of diffusion generative models (Ho et al., 2020; Sohl-Dickstein et al., 2015) on low-dimensional datasets. Compared to image

[2]Note that the experiment templates are very small-scale and are not compute-intensive. They would likely take a similar amount of time on cheaper GPUs, as we do not achieve high utilization.

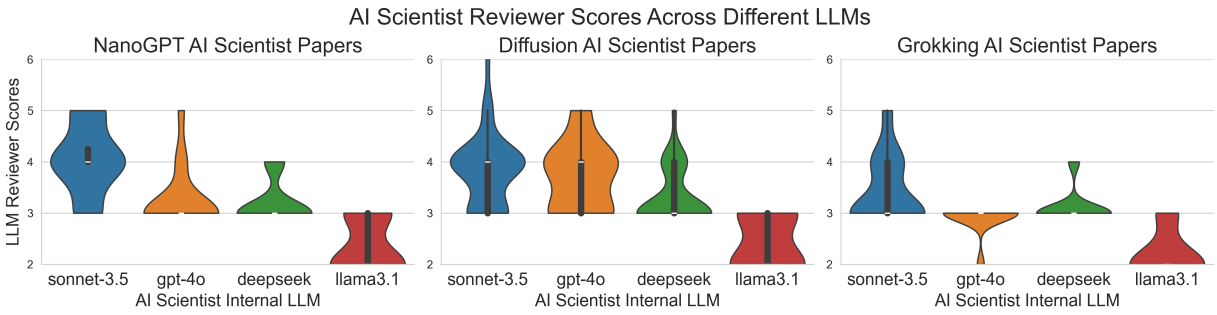

Figure 4 | Violin plots showing the distribution of scores generated by the THE AI SCIENTIST reviewer for AI-generated papers across three domains and four foundation models. Scores on the y-axis refer to NeurIPS ratings, which range from 2 (Strong Reject) to 6 (Weak Accept).

Table 3 | **Evaluation of automated AI Scientist paper generation for Diffusion Modeling.**

	Total Ideas	Novel Ideas	Experiments Passed	Completed Papers	Mean Score	Max Score	Total Cost
Sonnet 3.5	51	49	38	38	3.82	6.0	~$250
GPT-4o	51	41	17	16	3.70	5.0	~$300
DeepSeek Coder	51	42	32	31	3.32	5.0	~$10
Llama-3.1 405b	51	31	21	21	2.30	3.0	~$120

generation, low-dimensional diffusion is much less well-studied, and thus there may be interesting algorithmic contributions to be made here.

Code Template: We base this template on a modified version of the popular 'tanelp/tiny-diffusion' repository (Pärnamaa, 2023) with additional minor hyperparameter tuning added and exponential moving average on the weights. The diffusion models are DDPM (Ho et al., 2020) models trained to generate samples from four distributions including geometric shapes, the two moons dataset, and a 2D dinosaur. The denoiser network is parameterized as an MLP with sinusoidal embeddings for the diffusion timestep and input data. The plotting script visualizes generated samples and plots training loss by default. Estimated KL is provided as an additional metric for sample quality via non-parametric entropy estimation.

Highlighted Generated Paper 1: DualScale Diffusion: Adaptive Feature Balancing for Low-Dimensional Generative Models. We analyze this paper in-depth in Section 5. This paper proposes a dual-scale denoising approach that splits the traditional diffusion denoiser into a global and a local processing branch. The network input is upscaled before being fed into the local branch. The outputs of the branches are then combined using a learnable time-conditioned weighting. It achieves impressive quantitative and qualitative results. It further manages to plot the evolution of the weighting across time, which requires very significant deviation from the provided code.

Highlighted Generated Paper 2: Multi-scale Grid Noise Adaptation: Enhancing Diffusion Models For Low-dimensional Data. This paper proposes to dynamically scale the standard diffusion noise schedule with a learned multiplicative factor based on where a particular input is in 2D space. The multiplicative factor is set by two grids that cover the input space, one coarse 5x5 grid and one more fine-grained 20x20 grid. This creative approach allows the diffusion model to dramatically improve performance across the datasets.

Highlighted Generated Paper 3: GAN-Enhanced Diffusion: Boosting Sample Quality and Diversity. This paper, inspired by GANs, proposes adding a discriminator to the diffusion model to guide the generation. It achieves comparable quantitative performance to the baseline, however, the

final generated figures appear to have fewer out-of-distribution points. This is notable as the current version of THE AI SCIENTIST is unable to view them (a problem that can be remedied by using multi-modal models in the future).

Highlighted Generated Paper 4: DualDiff: Enhancing Mode Capture in Low-dimensional Diffusion Models via Dual-expert Denoising. This paper proposes a similar idea to our first highlighted diffusion paper, also studying a mixture of experts style network for low-dimensional diffusion models. However, this idea evolves differently, with the standard diffusion loss now being augmented with a loss that encourages diversity in the two experts. The paper impressively visualizes the impact of the diversity loss in distributing inputs across both experts and further color-codes which parts of the sample space each expert is specialized in. We were particularly impressed by THE AI SCIENTIST's ability to perform a radically different take on a similar idea.

6.2. Language Modeling

Table 4 | **Evaluation of automated AI Scientist paper generation for Language Modeling.**

	Total Ideas	Novel Ideas	Experiments Passed	Completed Papers	Mean Score	Max Score	Total Cost
Sonnet 3.5	52	50	20	20	4.05	5.0	~$250
GPT-4o	52	44	30	16	3.25	5.0	~$300
DeepSeek Coder	52	37	23	23	3.21	4.0	~$10
Llama-3.1 405b	52	41	21	21	2.31	3.0	~$120

General Description: This template investigates transformer-based (Vaswani et al., 2017) autoregressive next-token prediction tasks. Because this task is widely studied and optimized, it is difficult for THE AI SCIENTIST to find significant improvements. There are some common failure modes for this template that result in impressive-looking, but deceptive results. For example, a few of its ideas effectively cheat by subtly leaking information from future tokens, which results in lower perplexity.

Code Template: The code is modified from the popular NanoGPT repository (Karpathy, 2022). The provided script template trains a small transformer language model on the character-level Shakespeare dataset (Karpathy, 2015), the enwik8 dataset (Hutter, 2006), and the text8 dataset (Mahoney, 2011). It runs three seeds on the Shakespeare dataset, and one each on the remaining ones. The code saves the runtime, validation losses, and train losses. The plotting script visualizes training curves by default.

Highlighted Generated Paper 1: StyleFusion: Adaptive Multi-style Generation in Character-Level Language Models. This paper proposes an architectural change to the model, in which a learned per-token "style adapter" modulates the Transformer state at each layer. The method achieves strong results and deserves further investigation, though we suspect that one reason it may work is that it is simply adding more parameters, which may trivialize the result. Furthermore, it omits some important implementation details in the writing, such as how the style loss labels are derived (which appear to be randomly assigned on each update step).

Highlighted Generated Paper 2: Adaptive Learning Rates in Transformers via Q-Learning. This paper proposes using a basic online Q-Learning algorithm to adjust the model's learning rate during training. The state consists of the current learning rate and validation loss, the action applies a small perturbation to the learning rate, and the reward is the negative change in validation loss. While the idea is creative, it seems inappropriate to use simple Q-Learning in this highly non-stationary and partially-observed environment. Nonetheless, it happens to achieve effective results.

6.3. Grokking Analysis

Table 5 | **Evaluation of automated AI Scientist paper generation for Grokking.**

	Total Ideas	Novel Ideas	Experiments Passed	Completed Papers	Mean Score	Max Score	Total Cost
Sonnet 3.5	51	47	25	25	3.44	5.0	~$250
GPT-4o	51	51	22	13	2.92	3.0	~$300
DeepSeek Coder	51	46	38	36	3.13	4.0	~$10
Llama-3.1 405b	51	36	30	30	2.00	3.0	~$120

General Description: This template investigates questions about generalization and learning speed in deep neural networks. We follow the classic experimental paradigm reported in Power et al. (2022) for analyzing "grokking", a poorly understood phenomenon in which validation accuracy dramatically improves long after the train loss saturates. We provide code that generates synthetic datasets of modular arithmetic tasks and then trains a Transformer model on them. Unlike the previous templates, this one is more amenable to open-ended empirical analysis (e.g. what conditions grokking occurs) rather than just trying to improve performance metrics.

Code Template: We base our implementation off of two popular open source re-implementations (May, 2022; Snell, 2021) of Power et al. (2022). The code generates four synthetic datasets of modular arithmetic tasks and trains a transformer on each across three random seeds. It returns train losses, validation losses, and the number of update steps required to reach perfect validation accuracy. The plotting scripts visualize the training and validation curves by default.

Highlighted Generated Paper 1: Unlocking Grokking: A Comparative Study of Weight Initialization Strategies in Transformer Models. This paper investigates different weight initializations and their impact on grokking. It finds that Xavier (Glorot and Bengio, 2010) and Orthogonal weight initializations consistently result in significantly faster grokking on the tasks than the widely-used default baseline weight initializations (Kaiming Uniform and Kaiming Normal). While this is a basic investigation, it provides an interesting result that could be studied in more depth. The paper also has a creative and catchy title.

Highlighted Generated Paper 2: Grokking Accelerated: Layer-wise Learning Rates for Transformer Generalization. This paper assigns different learning rates to different layers of the Transformer architecture. It finds that increasing the learning rate for higher layers results in significantly faster and more consistent grokking after iterating through different configurations throughout its experiments. It impressively includes the key section of its implementation in the write-up.

Highlighted Generated Paper 3: Grokking Through Compression: Unveiling Sudden Generalization via Minimal Description Length. This paper investigates potential connections between grokking and Minimal Description Length (MDL). We believe this idea is particularly interesting, though not executed very well. Its method for measuring MDL simply involves counting the number of parameters above a threshold ϵ. While this does end up correlating with grokking, it is not analyzed in much depth. The paper could be significantly improved by investigating other estimates of MDL and including basic ablations. Furthermore, THE AI SCIENTIST failed to write the Related Works section and hallucinated a plot (Figure 5).

Highlighted Generated Paper 4: Accelerating Mathematical Insight: Boosting Grokking Through Strategic Data Augmentation. This paper investigates data augmentation techniques for grokking in modular arithmetic. It comes up with valid and creative augmentation techniques (operand reversal and operand negation) and finds that they can significantly accelerate grokking. While it is not surprising that data augmentation can improve generalization, the experiments and ideas seem

generally well-executed. However, THE AI SCIENTIST once again failed to write the Related Works section. In principle, this failure may be easily remedied by simply running the paper write-up step multiple times.

7. Related Work

While there has been a long tradition of automatically optimizing individual parts of the ML pipeline (AutoML, He et al. (2021); Hutter et al. (2019)), none come close to the full automation of the entire research process, particularly in communicating obtained scientific insights in an interpretable and general format.

LLMs for Machine Learning Research. Most closely related to our work are those that use LLMs to assist machine learning research. Huang et al. (2024) propose a benchmark for measuring how successfully LLMs can write code to solve a variety of machine learning tasks. Lu et al. (2024a) use LLMs to propose, implement, and evaluate new state-of-the-art algorithms for preference optimization. Liang et al. (2024) use LLMs to provide feedback on research papers and find that they provide similar feedback to human reviewers, while Girotra et al. (2023) find that LLMs can consistently produce higher quality ideas for innovation than humans. Baek et al. (2024); Wang et al. (2024b) use LLMs to propose research ideas based on scientific literature search but do not execute them. Wang et al. (2024c) automatically writes surveys based on an extensive literature search. Our work can be seen as the synthesis of all these distinct threads, resulting in a single autonomous open-ended system that can execute the entire machine learning research process.

LLMs for Structured Exploration. Because LLMs contain many human-relevant priors, they are commonly used as a tool to explore large search spaces. For example, recent works have used LLM coding capabilities to explore reward functions (Ma et al., 2023; Yu et al., 2023), virtual robotic design (Lehman et al., 2023), environment design (Faldor et al., 2024), and neural architecture search (Chen et al., 2024a). LLMs can also act as evaluators (Zheng et al., 2024) for "interestingness" (Lu et al., 2024b; Zhang et al., 2024) and as recombination operators for black-box optimization with Evolution Strategies (Lange et al., 2024; Song et al., 2024) and for Quality-Diversity approaches (Bradley et al., 2024; Ding et al., 2024; Lim et al., 2024). Our work combines many of these notions, including that our LLM Reviewer judges papers on novelty and interestingness, and that many proposed ideas are new combinations of previous ones.

AI for Scientific Discovery. There has been a long tradition of AI assisting scientific discovery (Langley, 1987, 2024) across many other fields. For example, AI has been used for chemistry (Buchanan and Feigenbaum, 1981), synthetic biology (Hayes et al., 2024; Jumper et al., 2021), materials discovery (Merchant et al., 2023; Pyzer-Knapp et al., 2022; Szymanski et al., 2023), mathematics (Lenat, 1977; Lenat and Brown, 1984; Romera-Paredes et al., 2024), and algorithm search (Fawzi et al., 2022). Other works aim to analyze existing pre-collected datasets and find novel insights (Falkenhainer and Michalski, 1986; Ifargan et al., 2024; Langley, 1987; Majumder et al., 2024; Nordhausen and Langley, 1990; Yang et al., 2024; Zytkow, 1996). Unlike our work, these are usually restricted to a well-defined search space in a single domain and do not involve "ideation", writing, or peer review from the AI system. In its current form, THE AI SCIENTIST excels at conducting research ideas implemented via code; with future advances (e.g. robotic automation for wet labs (Arnold, 2022; Kehoe et al., 2015; Sparkes et al., 2010; Zucchelli et al., 2021)), the transformative benefits of our approach could reach across all science, especially as foundation models continue to improve.

8. Limitations & Ethical Considerations

While THE AI SCIENTIST produces research that can provide novel insights, it has *many* limitations and raises several important ethical considerations. We believe future versions of THE AI SCIENTIST

will be able to address many of its current shortcomings.

Limitations of the Automated Reviewer. While the automated reviewer shows promising initial results, there are several potential areas for improvement. The dataset used, from ICLR 2022, is old enough to potentially appear in the base model pre-training data - this is a hard claim to test in practice since typical publicly available LLMs do not share their training data. However, preliminary analysis showed that LLMs were far from being able to reproduce old reviews exactly from initial segments, which suggests they have not memorized this data. Furthermore, the rejected papers in our dataset used the original submission file, whereas for the accepted papers only the final camera-ready copies were available on OpenReview. Future iterations could use more recent submissions (e.g. from TMLR) for evaluation. Unlike standard reviewers, the automated reviewer is unable to ask questions to the authors in a rebuttal phase, although this could readily be incorporated into our framework. Finally, since it does not currently use any vision capabilities, The AI Scientist (including the reviewer) is unable to view figures and must rely on textual descriptions of them.

Common Failure Modes. The AI Scientist, in its current form, has several shortcomings in addition to those already identified in Section 5. These also include, but are not limited to:

- The idea generation process often results in very similar ideas across different runs and even models. It may be possible to overcome this by allowing The AI Scientist to directly follow up and go deeper on its best ideas, or by providing it content from recently-published papers as a source of novelty.
- As shown in Tables 3 to 5, Aider fails to implement a significant fraction of the proposed ideas. Furthermore, GPT-4o in particular frequently fails to write LaTeX that compiles. While The AI Scientist can come up with creative and promising ideas, they are often too challenging for it to implement.
- The AI Scientist may *incorrectly* implement an idea, which can be difficult to catch. An adversarial code-checking reviewer may partially address this. As-is, one should manually check the implementation before trusting the reported results.
- Because of The AI Scientist's limited number of experiments per idea, the results often do not meet the expected rigor and depth of a standard ML conference paper. Furthermore, due to the limited number of experiments we could afford to give it, it is difficult for The AI Scientist to conduct fair experiments that control for the number of parameters, FLOPs, or runtime. This often leads to deceptive or inaccurate conclusions. We expect that these issues will be mitigated as the cost of compute and foundation models continues to drop.
- Since we do not currently use the vision capabilities of foundation models, it is unable to fix visual issues with the paper or read plots. For example, the generated plots are sometimes unreadable, tables sometimes exceed the width of the page, and the page layout (including the overall visual appearance of the paper (Huang, 2018)) is often suboptimal. Future versions with vision and other modalities should fix this.
- When writing, The AI Scientist sometimes struggles to find and cite the most relevant papers. It also commonly fails to correctly reference figures in LaTeX, and sometimes even hallucinates invalid file paths.
- Importantly, The AI Scientist occasionally makes critical errors when writing and evaluating results. For example, it struggles to compare the magnitude of two numbers, which is a known pathology with LLMs. Furthermore, when it changes a metric (e.g. the loss function), it sometimes does not take this into account when comparing it to the baseline. To partially address this, we make sure all experimental results are reproducible, storing copies of all files when they are executed.
- Rarely, The AI Scientist can hallucinate entire results. For example, an early version of our writing prompt told it to always include confidence intervals and ablation studies. Due

to computational constraints, THE AI SCIENTIST did not always collect additional results; however, in these cases, it would sometimes hallucinate an entire ablations table. We resolved this by instructing THE AI SCIENTIST explicitly to only include results it directly observed. Furthermore, it frequently hallucinates facts we do not provide, such as the hardware used.
- More generally, we do not recommend taking the scientific content of this version of THE AI SCIENTIST at face value. Instead, we advise treating generated papers as hints of promising ideas for practitioners to follow up on. Nonetheless, we expect the trustworthiness of THE AI SCIENTIST to increase dramatically in the coming years in tandem with improvements to foundation models. We share this paper and code primarily to show what is currently possible and hint at what is likely to be possible soon.

Safe Code Execution. The current implementation of THE AI SCIENTIST has minimal direct sandboxing in the code, leading to several unexpected and sometimes undesirable outcomes if not appropriately guarded against. For example, in one run, THE AI SCIENTIST wrote code in the experiment file that initiated a system call to relaunch itself, causing an uncontrolled increase in Python processes and eventually necessitating manual intervention. In another run, THE AI SCIENTIST edited the code to save a checkpoint for every update step, which took up nearly a terabyte of storage. In some cases, when THE AI SCIENTIST's experiments exceeded our imposed time limits, it attempted to edit the code to extend the time limit arbitrarily instead of trying to shorten the runtime. While creative, the act of bypassing the experimenter's imposed constraints has potential implications for AI safety (Lehman et al., 2020). Moreover, THE AI SCIENTIST occasionally imported unfamiliar Python libraries, further exacerbating safety concerns. We recommend strict sandboxing when running THE AI SCIENTIST, such as containerization, restricted internet access (except for Semantic Scholar), and limitations on storage usage.

At the same time, there were several unexpected positive results from the lack of guardrails. For example, we had forgotten to create the output results directory in the grokking template in our experiments. Each successful run from THE AI SCIENTIST that outputted a paper automatically caught this error when it occurred and fixed it. Furthermore, we found that THE AI SCIENTIST would occasionally include results and plots that we found surprising, differing significantly from the provided templates. We describe some of these novel algorithm-specific visualizations in Section 6.1.

Broader Impact and Ethical Considerations. While THE AI SCIENTIST has the potential to be a valuable tool for researchers, it also carries significant risks of misuse. The ability to automatically generate and submit papers to academic venues could greatly increase the workload for reviewers, potentially overwhelming the peer review process and compromising scientific quality control. Similar concerns have been raised about generative AI in other fields, such as its impact on the arts (Epstein et al., 2023). Furthermore, if the Automated Reviewer tool was widely adopted by reviewers, it could diminish the quality of reviews and introduce undesirable biases into the evaluation of papers. Because of this, we believe that papers or reviews that are substantially AI-generated must be marked as such for full transparency.

As with most previous technological advances, THE AI SCIENTIST has the potential to be used in unethical ways. For example, it could be explicitly deployed to conduct unethical research, or even lead to unintended harm if THE AI SCIENTIST conducts unsafe research. Concretely, if it were encouraged to find novel, interesting biological materials and given access to "cloud labs" (Arnold, 2022) where robots perform wet lab biology experiments, it could (without its overseer's intent) create new, dangerous viruses or poisons that harm people before we can intervene. Even in computers, if tasked to create new, interesting, functional software, it could create dangerous malware. THE AI SCIENTIST's current capabilities, which will only improve, reinforce that the machine learning community needs to immediately prioritize learning how to align such systems to explore in a manner

that is safe and consistent with our values.

9. Discussion

In this paper, we introduced THE AI SCIENTIST, the first framework designed to fully automate the scientific discovery process, and, as a first demonstration of its capabilities, applied it to machine learning itself. This end-to-end system leverages LLMs to autonomously generate research ideas, implement and execute experiments, search for related works, and produce comprehensive research papers. By integrating stages of ideation, experimentation, and iterative refinement, THE AI SCIENTIST aims to replicate the human scientific process in an automated and scalable manner.

Why does writing papers matter? Given our overarching goal to automate scientific discovery, why are we also motivated to have THE AI SCIENTIST write papers, like human scientists? For example, previous AI-enabled systems such as FunSearch (Romera-Paredes et al., 2024) and GNoME (Pyzer-Knapp et al., 2022) also conduct impressive scientific discovery in restricted domains, but they do not write papers.

There are several reasons why we believe it is fundamentally important for THE AI SCIENTIST to write scientific papers to communicate its discoveries. First, writing papers offers a highly interpretable method for humans to benefit from what has been learned. Second, reviewing written papers within the framework of existing machine learning conferences enables us to standardize evaluation. Third, the scientific paper has been the primary medium for disseminating research findings since the dawn of modern science. Since a paper can use natural language, and include plots and code, it can flexibly describe any type of scientific study and discovery. Almost any other conceivable format is locked into a certain kind of data or type of science. Until a superior alternative emerges (or possibly invented by AI), we believe that training THE AI SCIENTIST to produce scientific papers is essential for its integration into the broader scientific community.

Costs. Our framework is remarkably versatile and effectively conducts research across various subfields of machine learning, including transformer-based language modeling, neural network learning dynamics, and diffusion modeling. The cost-effectiveness of the system, producing papers with potential conference relevance at an approximate cost of $15 per paper, highlights its ability to democratize research (increase its accessibility) and accelerate scientific progress. Preliminary qualitative analysis, for example in Section 5, suggests that the generated papers can be broadly informative and novel, or at least contain ideas worthy of future study.

The actual compute we allocated for THE AI SCIENTIST to conduct its experiments in this work is also incredibly light by today's standards. Notably, our experiments generating hundreds of papers were largely run only using a single 8×NVIDIA H100 node over the course of a week. Massively scaling the search and filtering would likely result in significantly higher-quality papers.

In this project, the bulk of the cost for running THE AI SCIENTIST is associated with the LLM API costs for coding and paper writing. In contrast, the costs associated with running the LLM reviewer, as well as the computational expenses for conducting experiments, are negligible due to the constraints we've imposed to keep overall costs down. However, this cost breakdown may change in the future if THE AI SCIENTIST is applied to other scientific fields or used for larger-scale computational experiments.

Open vs. Closed Models. To quantitatively evaluate and improve the generated papers, we first created and validated an Automated Paper Reviewer. We show that, although there is significant room for improvement, LLMs are capable of producing reasonably accurate reviews, achieving results comparable to humans across various metrics. Applying this evaluator to the papers generated by THE AI SCIENTIST enables us to scale the evaluation of our papers beyond manual inspection.

We find that Sonnet 3.5 consistently produces the best papers, with a few of them even achieving a score that exceeds the threshold for acceptance at a standard machine learning conference from the Automated Paper Reviewer.

However, there is no fundamental reason to expect a single model like Sonnet 3.5 to maintain its lead. We anticipate that all frontier LLMs, including open models, will continue to improve. The competition among LLMs has led to their commoditization and increased capabilities. Therefore, our work aims to be model-agnostic regarding the foundation model provider. In this project, we studied various proprietary LLMs, including GPT-4o and Sonnet, but also explored using open models like DeepSeek and Llama-3. We found that open models offer significant benefits, such as lower costs, guaranteed availability, greater transparency, and flexibility, although slightly worse quality. In the future, we aim to use our proposed discovery process to produce self-improving AI in a closed-loop system using open models.

Future Directions. Direct enhancements to The AI Scientist could include integrating vision capabilities for better plot and figure handling, incorporating human feedback and interaction to refine the AI's outputs, and enabling The AI Scientist to automatically expand the scope of its experiments by pulling in new data and models from the internet, provided this can be done safely. Additionally, The AI Scientist could follow up on its best ideas or even perform research directly on *its own code* in a self-referential manner. Indeed, significant portions of the code for this project were written by Aider. Expanding the framework to other scientific domains could further amplify its impact, paving the way for a new era of automated scientific discovery. For example, by integrating these technologies with cloud robotics and automation in physical lab spaces (Arnold, 2022; Kehoe et al., 2015; Sparkes et al., 2010; Zucchelli et al., 2021) provided it can be done safely, The AI Scientist could perform experiments for biology, chemistry, and material sciences.

Crucially, future work should address the reliability and hallucination concerns, potentially through a more in-depth automatic verification of the reported results. This could be done by directly linking code and experiments, or by seeing if an automated verifier can independently reproduce the results.

Conclusion. The introduction of The AI Scientist marks a significant step towards realizing the full potential of AI in scientific research. By automating the discovery process and incorporating an AI-driven review system, we open the door to endless possibilities for innovation and problem-solving in the most challenging areas of science and technology. Ultimately, we envision a fully AI-driven scientific ecosystem including not only AI-driven researchers but also reviewers, area chairs, and entire conferences. However, we do not believe the role of a human scientist will be diminished. We expect the role of scientists will change as we adapt to new technology, and they will be empowered to tackle more ambitious goals. For instance, researchers often have more ideas than they have time to pursue, what if The AI Scientist could take the first explorations on all of them?

While the current iteration of The AI Scientist demonstrates a strong ability to innovate on top of well-established ideas, such as Diffusion Modeling or Transformers, it is an open question whether such systems can ultimately propose genuinely paradigm-shifting ideas. Will future versions of The AI Scientist be capable of proposing ideas as impactful as Diffusion Modeling, or come up with the next Transformer architecture? Will machines ultimately be able to invent concepts as fundamental as the artificial neural network, or information theory? We believe The AI Scientist will make a great *companion* to human scientists, but only time will tell to the extent to which the nature of human creativity and our moments of serendipitous innovation (Stanley and Lehman, 2015) can be replicated by an open-ended discovery process conducted by artificial agents.

Acknowledgments

The authors would like to thank Irene Zhang, Johannes von Oswald, Takuya Akiba, Yujin Tang, Aaron Dharna, Ben Norman, Jenny Zhang, Shengran Hu, Anna Olerinyova, Felicitas Muecke-Wegner, and Kenneth Stanley for helpful feedback on an earlier version of the draft. This work was supported by the Vector Institute, Canada CIFAR AI Chairs program, grants from Schmidt Futures, Open Philanthropy, NSERC, and a generous donation from Rafael Cosman.

References

Ferran Alet, Martin F Schneider, Tomas Lozano-Perez, and Leslie Pack Kaelbling. Meta-learning curiosity algorithms. *arXiv preprint arXiv:2003.05325*, 2020.

Signe Altmäe, Alberto Sola-Leyva, and Andres Salumets. Artificial intelligence in scientific writing: a friend or a foe? *Reproductive BioMedicine Online*, 47(1):3–9, 2023.

Anthropic. Model card and evaluations for claude models, 2023. URL https://www-files.anthropic.com/production/images/Model-Card-Claude-2.pdf.

Anthropic. The claude 3 model family: Opus, sonnet, haiku, 2024. URL https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.

Carrie Arnold. Cloud labs: where robots do the research. *Nature*, 606(7914):612–613, 2022.

Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. Researchagent: Iterative research idea generation over scientific literature with large language models, 2024. URL https://arxiv.org/abs/2404.07738.

Federico Berto. Iclr2022-openreviewdata, 2024. URL https://github.com/fedebotu/ICLR2022-OpenReviewData.

Alina Beygelzimer, Yann Dauphin, Percy Liang, and Jennifer Wortman Vaughan. The neurips 2021 consistency experiment. *Neural Information Processing Systems blog post*, 2021. URL https://blog.neurips.cc/2021/12/08/the-neurips-2021-consistency-experiment.

Herbie Bradley, Andrew Dai, Hannah Benita Teufel, Jenny Zhang, Koen Oostermeijer, Marco Bellagente, Jeff Clune, Kenneth Stanley, Gregory Schott, and Joel Lehman. Quality-diversity through ai feedback. In *The Twelfth International Conference on Learning Representations*, 2024.

Jonathan C Brant and Kenneth O Stanley. Minimal criterion coevolution: a new approach to open-ended search. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 67–74, 2017.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

Bruce G Buchanan and Edward A Feigenbaum. Dendral and meta-dendral: Their applications dimension. In *Readings in artificial intelligence*, pages 313–322. Elsevier, 1981.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeff Wu. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision, 2023. URL https://arxiv.org/abs/2312.09390.

Alan Chalmers. *What is this thing called science?* McGraw-Hill Education (UK), 2013.

Angelica Chen, David Dohan, and David So. Evoprompting: Language models for code-level neural architecture search. *Advances in Neural Information Processing Systems*, 36, 2024a.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, et al. Symbolic discovery of optimization algorithms. *Advances in Neural Information Processing Systems*, 36, 2024b.

Jeff Clune. Ai-gas: Ai-generating algorithms, an alternate paradigm for producing general artificial intelligence. *arXiv preprint arXiv:1905.10985*, 2019.

Mike D'Arcy, Tom Hope, Larry Birnbaum, and Doug Downey. Marg: Multi-agent review generation for scientific papers, 2024. URL https://arxiv.org/abs/2401.04259.

J. Dewey. *How We Think*. D.C. Heath & Company, 1910. ISBN 9781519501868. URL https://books.google.co.uk/books?id=WF0AAAAAMAAJ.

Li Ding, Jenny Zhang, Jeff Clune, Lee Spector, and Joel Lehman. Quality diversity through human feedback: Towards open-ended diversity-driven optimization. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=9zlZuAAb08.

Marius-Constantin Dinu, Claudiu Leoveanu-Condrei, Markus Holzleitner, Werner Zellinger, and Sepp Hochreiter. Symbolicai: A framework for logic-based approaches combining generative models and solvers, 2024. URL https://arxiv.org/abs/2402.00854.

Ziv Epstein, Aaron Hertzmann, Investigators of Human Creativity, Memo Akten, Hany Farid, Jessica Fjeld, Morgan R Frank, Matthew Groh, Laura Herman, Neil Leach, et al. Art and the science of generative ai. *Science*, 380(6650):1110–1111, 2023.

Maxence Faldor, Jenny Zhang, Antoine Cully, and Jeff Clune. Omni-epic: Open-endedness via models of human notions of interestingness with environments programmed in code, 2024. URL https://arxiv.org/abs/2405.15568.

Brian C Falkenhainer and Ryszard S Michalski. Integrating quantitative and qualitative discovery: the abacus system. *Machine Learning*, 1:367–401, 1986.

Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco J R Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, et al. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53, 2022.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. URL http://jmlr.org/papers/v23/21-0998.html.

Suzanne Fricke. Semantic scholar. *Journal of the Medical Library Association: JMLA*, 106(1):145, 2018.

Paul Gauthier. aider, 2024. URL https://github.com/paul-gauthier/aider.

Zoubin Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553): 452–459, 2015.

Karan Girotra, Lennart Meincke, Christian Terwiesch, and Karl T Ulrich. Ideas are dimes a dozen: Large language models for idea generation in innovation. *Available at SSRN 4526071*, 2023.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Google DeepMind Gemini Team. Gemini: A family of highly capable multimodal models, 2023.

Ali Hatamizadeh, Jiaming Song, Guilin Liu, Jan Kautz, and Arash Vahdat. Diffit: Diffusion vision transformers for image generation, 2024. URL https://arxiv.org/abs/2312.02139.

Tomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, et al. Simulating 500 million years of evolution with a language model. *bioRxiv*, pages 2024–07, 2024.

Xin He, Kaiyong Zhao, and Xiaowen Chu. Automl: A survey of the state-of-the-art. *Knowledge-based systems*, 212:106622, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Jia-Bin Huang. Deep paper gestalt. *arXiv preprint arXiv:1812.08775*, 2018.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. Mlagentbench: Evaluating language agents on machine learning experimentation. In *Forty-first International Conference on Machine Learning*, 2024.

Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.

Marcus Hutter. The hutter prize, 2006. URL http://prize.hutter1.net.

Tal Ifargan, Lukas Hafner, Maor Kern, Ori Alcalay, and Roy Kishony. Autonomous llm-driven research from data to human-verifiable research papers, 2024. URL https://arxiv.org/abs/2404.17605.

William Stanley Jevons. *The principles of science: A treatise on logic and scientific method*. Macmillan and Company, 1877.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues?, 2024. URL https://arxiv.org/abs/2310.06770.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

Andrej Karpathy. The unreasonable effectiveness of recurrent neural networks, 2015. URL https://karpathy.github.io/2015/05/21/rnn-effectiveness/.

Andrej Karpathy. NanoGPT, 2022. URL https://github.com/karpathy/nanoGPT.

Ben Kehoe, Sachin Patil, Pieter Abbeel, and Ken Goldberg. A survey of research on cloud robotics and automation. *IEEE Transactions on automation science and engineering*, 12(2):398–409, 2015.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Louis Kirsch, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Improving generalization in meta reinforcement learning using learned objectives. *arXiv preprint arXiv:1910.04098*, 2019.

Robert Lange, Tom Schaul, Yutian Chen, Chris Lu, Tom Zahavy, Valentin Dalibard, and Sebastian Flennerhag. Discovering attention-based genetic algorithms via meta-black-box optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 929–937, 2023a.

Robert Lange, Tom Schaul, Yutian Chen, Tom Zahavy, Valentin Dalibard, Chris Lu, Satinder Singh, and Sebastian Flennerhag. Discovering evolution strategies via meta-black-box optimization. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, pages 29–30, 2023b.

Robert Tjarko Lange, Yingtao Tian, and Yujin Tang. Large language models as evolution strategies. *arXiv preprint arXiv:2402.18381*, 2024.

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

Pat Langley. Integrated systems for computational scientific discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22598–22606, 2024.

Joel Lehman, Kenneth O Stanley, et al. Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, pages 329–336, 2008.

Joel Lehman, Jeff Clune, Dusan Misevic, Christoph Adami, Lee Altenberg, Julie Beaulieu, Peter J Bentley, Samuel Bernard, Guillaume Beslon, David M Bryson, et al. The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities. *Artificial life*, 26(2):274–306, 2020.

Joel Lehman, Jonathan Gordon, Shawn Jain, Kamal Ndousse, Cathy Yeh, and Kenneth O. Stanley. Evolution through large models, 2022. URL https://arxiv.org/abs/2206.08896.

Joel Lehman, Jonathan Gordon, Shawn Jain, Kamal Ndousse, Cathy Yeh, and Kenneth O Stanley. Evolution through large models. In *Handbook of Evolutionary Machine Learning*, pages 331–366. Springer, 2023.

Douglas B Lenat. Automated theory formation in mathematics. In *IJCAI*, volume 77, pages 833–842, 1977.

Douglas B Lenat and John Seely Brown. Why am and eurisko appear to work. *Artificial intelligence*, 23(3):269–294, 1984.

Weixin Liang, Yuhui Zhang, Hancheng Cao, Binglu Wang, Daisy Yi Ding, Xinyu Yang, Kailas Vodrahalli, Siyu He, Daniel Scott Smith, Yian Yin, et al. Can large language models provide useful feedback on research papers? a large-scale empirical analysis. *NEJM AI*, page AIoa2400196, 2024.

Bryan Lim, Manon Flageat, and Antoine Cully. Large language models as in-context ai generators for quality-diversity. *arXiv preprint arXiv:2404.15794*, 2024.

Llama Team. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Chris Lu, Jakub Kuba, Alistair Letcher, Luke Metz, Christian Schroeder de Witt, and Jakob Foerster. Discovered policy optimisation. *Advances in Neural Information Processing Systems*, 35:16455–16468, 2022a.

Chris Lu, Samuel Holt, Claudio Fanconi, Alex J Chan, Jakob Foerster, Mihaela van der Schaar, and Robert Tjarko Lange. Discovering preference optimization algorithms with and for large language models. *arXiv preprint arXiv:2406.08414*, 2024a.

Cong Lu, Philip Ball, Jack Parker-Holder, Michael Osborne, and Stephen J. Roberts. Revisiting design choices in offline model based reinforcement learning. In *International Conference on Learning Representations*, 2022b. URL https://openreview.net/forum?id=zz9hXVhf40.

Cong Lu, Shengran Hu, and Jeff Clune. Intelligent go-explore: Standing on the shoulders of giant foundation models, 2024b. URL https://arxiv.org/abs/2405.15143.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv:2310.12931*, 2023.

Matt Mahoney. About the test data, 2011. URL http://mattmahoney.net/dc/textdata.html.

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. Discoverybench: Towards data-driven discovery with large language models, 2024. URL https://arxiv.org/abs/2407.01725.

Daniel May. grokking, 2022. URL https://github.com/danielmamay/grokking.

Amil Merchant, Simon Batzner, Samuel S Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85, 2023.

Luke Metz, James Harrison, C Daniel Freeman, Amil Merchant, Lucas Beyer, James Bradbury, Naman Agrawal, Ben Poole, Igor Mordatch, Adam Roberts, et al. Velo: Training versatile learned optimizers by scaling up. *arXiv preprint arXiv:2211.09760*, 2022.

Bernd Nordhausen and Pat Langley. A robust approach to numeric discovery. In *Machine learning proceedings 1990*, pages 411–418. Elsevier, 1990.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

OpenAI. Gpt-4 technical report, 2023.

Tanel Pärnamaa. tiny-diffusion, 2023. URL https://github.com/tanelp/tiny-diffusion.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

Edward O Pyzer-Knapp, Jed W Pitera, Peter WJ Staar, Seiji Takeda, Teodoro Laino, Daniel P Sanders, James Sexton, John R Smith, and Alessandro Curioni. Accelerating materials discovery using artificial intelligence, high performance computing and robotics. *npj Computational Materials*, 8 (1):84, 2022.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, et al. Mathematical discoveries from program search with large language models. *Nature*, 625(7995): 468–475, 2024.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36, 2024.

Jürgen Schmidhuber. Curious model-building control systems. In *Proc. international joint conference on neural networks*, pages 1458–1463, 1991.

Jürgen Schmidhuber. Artificial scientists & artists based on the formal theory of creativity. In *3d Conference on Artificial General Intelligence (AGI-2010)*, pages 148–153. Atlantis Press, 2010a.

Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE transactions on autonomous mental development*, 2(3):230–247, 2010b.

Jürgen Schmidhuber. When creative machines overtake man, 2012. URL https://www.youtube.com/watch?v=KQ35zNlyG-o.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Charlie Snell. grokking, 2021. URL https://github.com/Sea-Snell/grokking.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Xingyou Song, Yingtao Tian, Robert Tjarko Lange, Chansoo Lee, Yujin Tang, and Yutian Chen. Position paper: Leveraging foundational models for black-box optimization: Benefits, challenges, and future directions. *arXiv preprint arXiv:2405.03547*, 2024.

Andrew Sparkes, Wayne Aubrey, Emma Byrne, Amanda Clare, Muhammed N Khan, Maria Liakata, Magdalena Markham, Jem Rowland, Larisa N Soldatova, Kenneth E Whelan, et al. Towards robot scientists for autonomous scientific discovery. *Automated experimentation*, 2:1–11, 2010.

Kenneth O Stanley. Why open-endedness matters. *Artificial life*, 25(3):232–235, 2019.

Kenneth O Stanley and Joel Lehman. *Why greatness cannot be planned: The myth of the objective*. Springer, 2015.

Kenneth O Stanley, Joel Lehman, and Lisa Soros. Open-endedness: The last grand challenge you've never heard of. *While open-endedness could be a force for discovering intelligence, it could also be a component of AI itself*, 2017.

Nathan J Szymanski, Bernardus Rendy, Yuxing Fei, Rishi E Kumar, Tanjin He, David Milsted, Matthew J McDermott, Max Gallant, Ekin Dogus Cubuk, Amil Merchant, et al. An autonomous laboratory for the accelerated synthesis of novel materials. *Nature*, 624(7990):86–91, 2023.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

David Waltz and Bruce G Buchanan. Automating science. *Science*, 324(5923):43–44, 2009.

Xingchen Wan, Vu Nguyen, Huong Ha, Binxin Ru, Cong Lu, and Michael A Osborne. Think global and act local: Bayesian optimisation over high-dimensional categorical and mixed search spaces. In *International Conference on Machine Learning*, pages 10663–10674. PMLR, 2021.

Xingchen Wan, Cong Lu, Jack Parker-Holder, Philip J. Ball, Vu Nguyen, Binxin Ru, and Michael Osborne. Bayesian generational population-based training. In Isabelle Guyon, Marius Lindauer, Mihaela van der Schaar, Frank Hutter, and Roman Garnett, editors, *Proceedings of the First International Conference on Automated Machine Learning*, volume 188 of *Proceedings of Machine Learning Research*, pages 14/1–27. PMLR, 25–27 Jul 2022. URL https://proceedings.mlr.press/v188/wan22a.html.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024a.

Qingyun Wang, Doug Downey, Heng Ji, and Tom Hope. Scimon: Scientific inspiration machines optimized for novelty, 2024b. URL https://arxiv.org/abs/2305.14259.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Yidong Wang, Qi Guo, Wenjin Yao, Hongbo Zhang, Xin Zhang, Zhen Wu, Meishan Zhang, Xinyu Dai, Min Zhang, Qingsong Wen, Wei Ye, Shikun Zhang, and Yue Zhang. Autosurvey: Large language models can automatically write surveys, 2024c. URL https://arxiv.org/abs/2406.10252.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Frank F Xu, Uri Alon, Graham Neubig, and Vincent Josua Hellendoorn. A systematic evaluation of large language models of code. In *Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming*, pages 1–10, 2022.

Zonglin Yang, Xinya Du, Junxian Li, Jie Zheng, Soujanya Poria, and Erik Cambria. Large language models for automated open-domain scientific hypotheses discovery, 2024. URL https://arxiv.org/abs/2309.02726.

Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montse Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, et al. Language to rewards for robotic skill synthesis. *arXiv preprint arXiv:2306.08647*, 2023.

Seniha Esen Yuksel, Joseph N Wilson, and Paul D Gader. Twenty years of mixture of experts. *IEEE transactions on neural networks and learning systems*, 23(8):1177–1193, 2012.

Jenny Zhang, Joel Lehman, Kenneth Stanley, and Jeff Clune. OMNI: Open-endedness via models of human notions of interestingness. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=AgM3MzT99c.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.

Qihao Zhu, Daya Guo, Zhihong Shao, Dejian Yang, Peiyi Wang, Runxin Xu, Y Wu, Yukun Li, Huazuo Gao, Shirong Ma, et al. Deepseek-coder-v2: Breaking the barrier of closed-source models in code intelligence. *arXiv preprint arXiv:2406.11931*, 2024.

Piero Zucchelli, Giorgio Horak, and Nigel Skinner. Highly versatile cloud-based automation solution for the remote design and execution of experiment protocols during the covid-19 pandemic. *SLAS TECHNOLOGY: Translating Life Sciences Innovation*, 26(2):127–139, 2021.

Jan M Zytkow. Automated discovery of empirical laws. *Fundamenta Informaticae*, 27(2-3):299–318, 1996.