# BigBang: Pursuing Open-Ended Intelligence through Self-Evolving Synthesis of Verifiable Frontier Tasks

**The BigBang Team**

## Abstract

As Large Language Models (LLMs) approach human-expert performance, their continued development is increasingly constrained by training tasks conceived within the limits of human knowledge. We argue that open-ended capability growth requires verifiable frontier tasks: problems at the boundary of current knowledge whose candidate solutions can be objectively assessed through formal methods, computation, simulation, or domain-specific tools. To this end, we introduce BigBang, a general-purpose LLM that explores how intelligence can continue to develop beyond the limits of human-curated tasks. Starting from Qwen3.6-35B-A3B, BigBang is trained on verifiable frontier tasks generated by an adversarial, self-evolving synthesis framework. The framework comprises two core components: generator agents that continually propose and solve increasingly challenging scientific and technical problems, and critic agents that evaluate their correctness, verifiability, difficulty, scalability, and diversity. Held-out real research tasks further calibrate the critic and guide the evolving distribution of synthetic data. Through iterative generator–critic interaction, the synthesis process progressively improves its own task-generation and evaluation strategies, constituting an early data-level form of recursive self-improvement. BigBang substantially outperforms its base model across scientific research, reasoning, coding, and tool-use benchmarks, achieving aggregate performance between DeepSeek V4 Flash (284B) and DeepSeek V4 Pro (1.6T). These results suggest that the self-evolving synthesis of verifiable frontier tasks offers a promising path toward sustained capability growth and open-ended intelligence.

**Code** https://github.com/endless-frontier/BigBang-v1
**Model** https://huggingface.co/endless-frontier/BigBang-v1

## 1 Introduction

Artificial intelligence (AI) is advancing at unprecedented speed from a conversational tool, to a coding assistant, and potentially, in the future, a core engine of scientific discovery (Gottweis et al., 2026; Novikov et al., 2025). To advance human civilization, AI must not merely imitate what humans already know and can do; its intelligence must have the capacity to advance beyond the limits of human knowledge and capability. As AI systems approach and increasingly surpass human expert performance across a growing range of domains, a deeper limitation emerges: their training remains confined to tasks, problems, and objectives conceived within the bounds of human cognition (Maslej et al., 2025; Phan et al., 2025a). What kinds of tasks can expand a model's capabilities beyond the limits of human supervision? How can we create a scalable, self-sustaining source of high-quality training data that enables models to improve continuously? These questions are inseparable. The first asks what kind of training task can provide an open-ended source of capability growth; the second asks how such tasks can be generated and validated at the scale required for sustained improvement.

Our answer to the first question is verifiable frontier tasks. By frontier, we mean problems at the boundary of human knowledge, including those for which no known optimum exists. By verifiable, we mean that


Figure 1: BigBang-V1 on eight representative benchmarks spanning long-horizon search, software engineering, scientific research, and AI research. BigBang-V1 obtains the highest reported score among the selected 35B models on all eight benchmarks. It even exceeds DeepSeek V4 Pro Preview (1.6T) on FrontierScience Research, Humanity's Last Exam, PaperBench(Code-Dev) and BioMysteryBench-HD.

candidate solutions can be assessed using objective rules, formal methods, computational tools, simulators, or experimental feedback. Frontier problems offer an open-ended horizon for capability growth, while verifiability supplies accurate, stable, and scalable learning signals. Neither property is sufficient on its own. Frontier problems without reliable verification provide no trustworthy basis for optimization, whereas verifiable tasks without frontier-level difficulty impose a ceiling that capable models can rapidly approach. At their intersection, open-ended exploration becomes amenable to systematic training, reliable evaluation, and iterative improvement (Zhao et al., 2025; Novikov et al., 2025).

Scientific research is a natural and consequential source of verifiable frontier tasks. Its open-ended nature continually generates new questions, while many scientific problems admit objective feedback through formal analysis, computation, simulation, and domain-specific tools (de Moura and Ullrich, 2021; Roy and Oberkampf, 2011; Jumper et al., 2021). We therefore use science not to specialize a general-purpose

model, but as a demanding training ground for intelligence. Solving scientific problems requires rigorous reasoning, tool use, hypothesis testing, and long-horizon problem solving, capabilities that can transfer well beyond science. Scientific frontiers thus provide models with an enduring source of challenges, while increasingly capable models can in turn extend the reach of scientific inquiry.

If verifiable frontier tasks answer the question of what models should learn from, a self-evolving synthetic data pipeline answers the question of how those tasks can be continuously generated, validated, and scaled. Frontier training requires data that remains reliable, challenging, scalable, and diverse as model capabilities improve. While human expertise remains essential for identifying consequential problems, setting research directions, and providing high-value priors, as tasks approach the frontier, the pool of experts capable of continuously generating, solving, and rigorously evaluating them rapidly narrows. Human-generated data alone therefore cannot keep pace with model development. Models must increasingly participate in producing and refining their own training problems, while humans focus on setting objectives, designing environments, and overseeing critical stages.

Synthetic generation alone is not sufficient. A static generator eventually reaches its own difficulty ceiling and reproduces its systematic biases (Du et al., 2026c,b). The data pipeline must therefore evolve with the model by identifying current capability gaps, generating harder and more diverse problems, filtering spurious difficulty and unreliable feedback, and using performance on real tasks to calibrate subsequent rounds of production. (Du et al., 2026a) From this perspective, long-term model improvement depends not only on the amount of data consumed in a single training run, but also on the rate at which the data engine can discover new challenges, incorporate external feedback, and raise its own frontier. The evolution rate of the data pipeline may therefore become a key determinant of the evolution rate of the model itself.

Motivated by this view, we introduce BigBang, a general-purpose Large Language Model designed to pursue open-ended capability growth through autonomous scientific research. Initialized from Qwen3.6-35B-A3B (Qwen Team, 2026), BigBang is developed through efficient post-training on self-evolving, verifiable frontier tasks. The defining feature of BigBang's training is not a particular collection of existing scientific data. Instead, it is an adversarial, self-evolving synthetic data framework grounded in verifiable frontier research tasks. Generator agents continually propose and solve scientific problems, seeking tasks that are increasingly challenging yet tractable. Critic agents adversarially evaluate each proposal for correctness, difficulty, scalability, and diversity; rejected tasks are revised or replaced, while accepted tasks raise the target difficulty for subsequent generation rounds. To ground this process in real scientific value, critic agents compare the generated tasks against a held-out collection of real research tasks. This reference set anchors their assessment of what constitutes a meaningful frontier problem, helping them identify gaps in domain coverage and calibrate the difficulty and distribution of subsequent synthetic-data batches. Through this repeated generator–critic interaction, the system progressively expands the frontier of its synthetic training data. Together, these interacting loops form the BigBang training flywheel, which produces training data targeted at the evolving structure of real scientific research rather than relying solely on a fixed scientific corpus.

Using this framework, we synthesize high-difficulty post-training data spanning multiple scientific and technical domains. Training on these data yields substantial capability gains: BigBang outperforms its Qwen3.6-35B-A3B base model across a broad suite of benchmarks, with aggregate performance between DeepSeek V4 Flash and DeepSeek V4 Pro. It is particularly strong on scientific research tasks, while also achieving broad improvements in reasoning, coding, tool use, and other non-scientific domains.

Our results point to a scalable path toward sustained capability growth: verifiable frontier tasks set the direction of improvement, while adversarial, self-evolving data supplies its fuel. All post-training data for BigBang are synthesized by AI systems, making data production itself a form of AI4AI. While many AI4AI efforts center on coding, we use general scientific research as the substrate for improvement. Scientific research integrates searching, reading, ideation, computation, experimentation and writing, and therefore exercises a broader range of capabilities than coding alone. At its limit, AI4AI becomes recursive self-

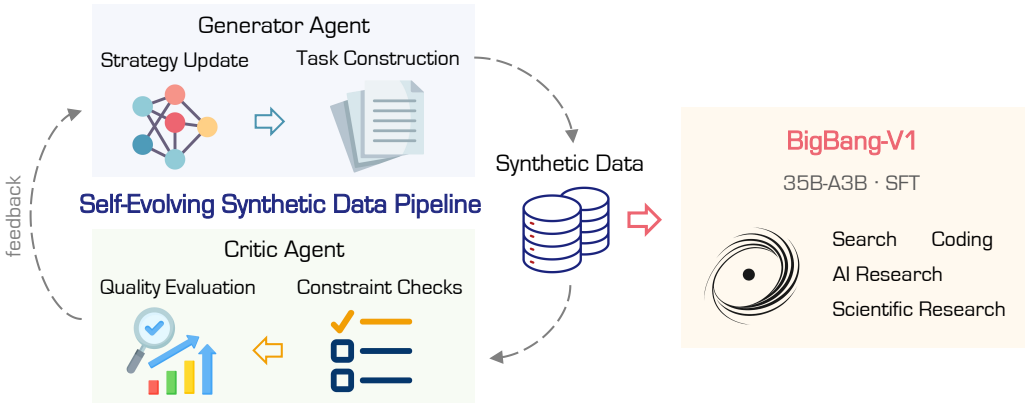


Figure 2: Overview of BigBang. A self-evolving synthetic data pipeline uses generator and critic agents to iteratively construct, evaluate, and refine verifiable tasks, producing high-quality synthetic data for post-training BigBang-V1.

improvement (RSI), as the outputs of one improvement cycle shape and strengthen the next. BigBang realizes an early, data-level form of this process: generator and critic agents repeatedly create, challenge, and refine the training data that drive subsequent capability gains. Note that science gives this loop room to grow. Its frontier continually yields new problems, while its methods provide a general framework for understanding and acting on the world. Science is therefore both an open-ended source of challenge and a general curriculum for intelligence. BigBang is an early step toward turning this reciprocal process, *"AI advancing science, and science advancing AI"*, into a practical path for the continued development of general intelligence.

## 2 Self-Evolving Synthesis of Verifiable Tasks

This section presents our self-evolving synthesis of verifiable tasks, comprising two tightly coupled components. The first is a self-evolving data synthesis pipeline, in which agents directly participate in the generation and evaluation of training data. The second is the co-evolution of the model and the data synthesis pipeline. The synthesized data are used to train the model, whose performance exposes current capability gaps and directs subsequent rounds of data synthesis for more challenging and diverse problems.

### 2.1 System Overview

**Self-evolving data synthesis pipeline.** We employ generator and critic agents to iteratively improve the synthesis of training data. The generator continually proposes and solves increasingly challenging candidate tasks, while the critic challenges the generated results according to their correctness, verifiability, difficulty, and training value. Candidates that fail the review are rejected or revised, whereas promising tasks inform subsequent rounds of generation. Through repeated adversarial interaction and mutual refinement, the pipeline progressively advances the difficulty and quality frontier of the synthesized data.

In the co-evolution of the model and the data synthesis pipeline, model performance on real tasks provides continuous feedback for subsequent rounds of data production. The system uses evaluations on real tasks to identify current capability gaps and accordingly adjusts the domain distribution, difficulty, and capability requirements of future tasks, driving the Generator agents to produce increasingly challenging and diverse candidates. Meanwhile, these evaluation results are also used to calibrate the critic agents' judgments of task quality and training value, enabling them to more accurately filter out spurious difficulty, unreliable feedback, and opportunistic examples that exploit weaknesses in the verification or review

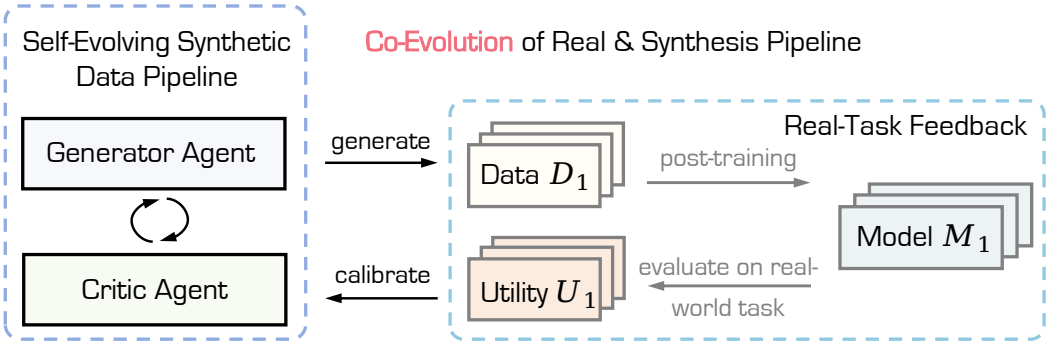


Figure 3: Our two-level co-evolution framework bridges synthetic data generation and real-task utility. The generator–critic loop synthesizes and filters verifiable data, while held-out real-world evaluation calibrates the critic and guides the next round of synthesis, reducing the synthetic-to-real gap.

process. The filtered data are then used to train new model variants, whose updated performance further guides the next round of generation and review. In this way, real frontier tasks provide the direction of evolution, verifiability supplies trustworthy feedback, and repeated generator-critic interaction enables scalable data production. Together, these components form the BigBang training flywheel: real frontier tasks $\rightarrow$ capability-gap identification $\rightarrow$ generator-critic interaction $\rightarrow$ high-quality synthetic data $\rightarrow$ model training $\rightarrow$ validation on real tasks $\rightarrow$ the next round of evolution.

## 2.2 Self-Evolution of Synthesis Pipeline

BigBang builds a self-evolving data synthesis system consisting of a generator and a critic. The generator continuously adjusts its strategies for task construction, problem solving, result verification, and sample selection, while the critic evaluates the output of each round and provides feedback. Ideally, a synthesis strategy should be assessed by training a model and measuring changes in downstream capabilities. However, repeated training and evaluation are costly and time-consuming, making it difficult to obtain timely feedback during large-scale strategy search. We therefore use the critic as a fast proxy for the training value of synthetic data. It provides fine-grained and interpretable optimization signals to the generator without requiring frequent model training.

**Generator.** The generator operates as a code agent that can directly modify, execute, and debug the data synthesis code, then refine its generation strategy based on feedback from the critic. This process can be viewed as a continuous search over programs, where choices about task construction, tool use, verification, and candidate filtering define a broad range of possible synthesis strategies. To reduce unproductive exploration, we initialize the search in a program region that can already generate promising scientific questions, allowing the generator to focus on improving strategies with greater potential. We also design an agent harness for long-term exploration. After each experiment, the generator records the motivation for its changes, experimental results, causes of failure, and subsequent conclusions. These records allow later rounds to inherit and reuse experience from earlier searches.

**Critic.** The critic evaluates candidate data at two levels: basic constraints and higher-level evidence of quality. Basic constraints examine whether the reasoning and interaction traces contain enough information, whether tools are used effectively, whether the data format is complete, and whether each sample can be correctly parsed by the training and execution systems. These checks do not directly prove that the data has high training value, but they can quickly remove clearly invalid or degraded outputs. The critic then determines whether key conclusions are supported by evidence, whether different steps contain factual or logical contradictions, whether the final answer can be objectively verified, and whether the task

requires non-trivial research and reasoning. The first level ensures basic validity, while the second assesses correctness, difficulty, and research value.

Through this closed loop, the generator receives targeted signals for improving specific parts of the pipeline without waiting for full model training, which substantially increases the efficiency of strategy search. During the experiments, the generator gradually develops several behaviors that were not explicitly specified. These include combining multiple information sources to deepen reasoning, revising generation prompts to establish clearer task boundaries, and adding quality checks and candidate elimination mechanisms to the synthesis process. This shows that the system optimizes not only the final samples but also the data production process itself. With the critic held fixed, the data produced by the generator shows consistent quality improvements on an independent benchmark across successive rounds. These results indicate that dense proxy feedback can effectively guide the continued evolution of the synthesis pipeline.

### 2.3 Co-Evolution of Model and Synthesis Pipeline

**Static pipeline saturation.** As model capabilities improve, a fixed synthesis pipeline gradually loses its ability to produce data that meaningfully extend the model's capability boundary. Previously challenging tasks become less informative, while fixed generation and evaluation strategies may repeat familiar patterns or favor tasks that appear difficult but are poorly verified and offer limited training value. The synthesis pipeline must therefore evolve with the model. It should continually identify emerging capability gaps and adjust the difficulty, diversity, and structure of the generated data.

**Outcome calibrated co-evolution.** BigBang anchors this evolution in a set of verifiable frontier tasks. We periodically select representative synthesis pipelines and use their generated data to train different model variants. We then evaluate these models on held-out real-world tasks, using their performance as an empirical measure of data utility. To incorporate this signal into subsequent data production, we introduce a meta-critic that compares the critic's assessments with the observed training outcomes. The discrepancy is then used to refine both evaluation criteria and generation strategies. This calibration helps distinguish tasks that merely appear difficult from those that genuinely improve model capabilities. Overall, the feedback loop keeps the synthesis pipeline aligned with the model's evolving capabilities and sustains continued model improvement.

## 3 Experimental Results and Case Studies

### 3.1 Experimental Setup

**Implementation.** We instantiate BigBang-V1 from Qwen3.6-35B-A3B (Qwen Team, 2026), which has 35B total parameters and 3B activated parameters during inference. The agent uses a 256k context window and allows up to 500 tool calls per trajectory.

**Benchmarks.** We evaluate BigBang-V1 on nine benchmark suites covering long-horizon search, coding, scientific research, and AI research. For long-horizon search, BrowseComp (Wei et al., 2025) tests the ability to locate difficult, entangled facts on the open web, while xbench-DeepSearch (Xbench-Team, 2025) evaluates the full planning–search–reasoning–summarization process. The coding group consists of SWE-Bench Pro (Deng et al., 2025), evaluated with the mini-SWE-Agent harness (Yang et al., 2024), and SciCode-Verified (Hu et al., 2026), our audited and corrected version of SciCode (Tian et al., 2024). SWE-Bench Pro uses realistic repository-level software-engineering issues, whereas SciCode-Verified measures scientific code generation at both the decomposed subproblem level (SciCode-V-Sub) and the end-to-end problem level (SciCode-V-Main). To construct SciCode-Verified, we audit all 65 original test problems, correct confirmed defects in their specifications and grading, and exclude the single problem for which no verifiable gold answer can be defined.

Table 1: Comparison of BigBang-V1 with representative closed- and open-source frontier models, together with models at the 35B scale, across benchmarks for long-horizon search, coding, scientific research, and AI research. The "−" indicates the score is not publicly available or not tested.

| Benchmark | *Closed-Source Models* | | | *Open-Source Models* | | | | *35B Models* | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Claude Opus 4.8 | Gemini 3.1 Pro | GPT 5.5 | GLM 5.2 | DeepSeek V4 Flash Preview | DeepSeek V4 Pro Preview | Step-3.7 Flash | Qwen3.6 35B-A3B | Nex-N2 mini | Agents A1 | Apodex 1.0-mini | BigBang V1 |
| *Long-horizon Search* | | | | | | | | | | | | |
| **BrowseComp** | 84.3 | 85.9 | 84.4 | 68.7 | 73.2 | 83.4 | 75.8 | 67.9 | 74.1 | 48.5 | 73.9 | **76.5** |
| **xbench** | 61.4 | 65.0 | 72.4 | 65.8 | 62.2 | 64.8 | 50.8 | 32.6 | 57.2 | 52.4 | 61.8 | **58.4** |
| *Coding Tasks* | | | | | | | | | | | | |
| **SWE-Bench Pro** | 69.2 | 54.2 | 58.6 | 62.1 | 52.6 | 55.4 | 56.3 | 43.6 | 50.2 | 42.3 | 38.7 | **54.2** |
| **SciCode-V-Sub** | 92.3 | - | 95.1 | 84.3 | 83.7 | 90.2 | - | 56.5 | 39.0 | 64.1 | - | **68.6** |
| **SciCode-V-Main** | 78.1 | - | 90.6 | 70.3 | 68.6 | 78.1 | - | 26.6 | 15.6 | 50.0 | - | **50.0** |
| *Scientific Research* | | | | | | | | | | | | |
| **FS-R** | 45.2 | 24.8 | 58.3 | 52.4 | 37.7 | 40.7 | 37.2 | 11.9 | 36.8 | 38.4 | 29.6 | **46.2** |
| **HLE** | 57.9 | 51.4 | 52.2 | 54.7 | 45.1 | 48.2 | 47.2 | 36.2 | 38.4 | 46.3 | 45.3 | **50.3** |
| **BioMystery-HS** | 88.5 | - | 76.7 | 75.3 | 68.0 | 64.4 | 57.5 | 44.8 | 42.9 | 48.9 | 50.2 | **57.5** |
| **BioMystery-HD** | 42.4 | - | 23.5 | 21.6 | 23.5 | 13.7 | 11.8 | 2.0 | 5.9 | 2.0 | 5.9 | **15.7** |
| *AI Research* | | | | | | | | | | | | |
| **MLE-Bench(Lite)** | 63.6 | - | 59.1 | 72.7 | 40.9 | 59.1 | 40.9 | 31.8 | 18.2 | 27.3 | 27.3 | **59.1** |
| **PaperBench(Code-Dev)** | - | - | 64.2 | 63.6 | 40.4 | 50.4 | 36.7 | 30.7 | 14.8 | 17.3 | 20.5 | **53.6** |

The scientific-research group includes FrontierScience-Research (FS-R, developed by OpenAI) (Wang et al., 2026), Humanity's Last Exam (HLE, developed by Scale AI) (Phan et al., 2025b), and BioMysteryBench (developed by Claude) (Anthropic, 2026b). These benchmarks cover open-ended scientific reasoning, expert-level knowledge, and tool-using bioinformatics. We report BioMysteryBench separately on its human-solvable (HS) and human-difficult (HD) subsets. The AI-research group contains MLE-Bench (developed by OpenAI) (Chan et al., 2025), which evaluates agents on competition-style machine-learning engineering, and PaperBench (developed by OpenAI) (Starace et al., 2025), which evaluates end-to-end replication of machine-learning papers. We report MLE-Bench(Lite) and PaperBench(Code-Dev) results here. The scientific group measures the capabilities targeted directly by BigBang's data pipeline; the other three groups measure transfer to web search, general software engineering, and autonomous AI research.

**Baselines.** Table 1 separates the evaluated models into three groups. The *Closed-Source Models* group contains Claude Opus 4.8 (Anthropic, 2026a), Gemini 3.1 Pro (Google DeepMind, 2026), and GPT-5.5 (OpenAI, 2026). The *Open-Source Models* group contains GLM-5.2 (Z.ai, 2026), DeepSeek V4 Flash Preview and Pro Preview (DeepSeek Team, 2026), and Step-3.7 Flash (StepFun, 2026). Together, these groups compare BigBang-V1 with frontier closed-source and open-source models. The *35B Models* group compares BigBang-V1 with its Qwen3.6-35B-A3B backbone (Qwen Team, 2026), Nex-N2-mini (Nex AGI, 2026), Agents-A1 (Bai et al., 2026), and Apodex-1.0-mini (Apodex Team, 2026). Qwen3.6-35B-A3B provides the direct reference for measuring the effect of BigBang's post-training, while the remaining 35B systems establish the comparison at the same model scale.

**Evaluation protocol.** Except for benchmarks with a dedicated harness, the evaluation environment exposes `search`, `visit`, and `code_exec` under the context and tool-call budgets specified above. BrowseComp additionally permits at most five discard_all operations per trajectory, and access to Hugging Face domains is blocked to prevent direct retrieval of benchmark artifacts or reference answers. To match their official default settings, we report FS-R with `avg@30` and xbench with `avg@5`; the remaining benchmarks follow their task-specific evaluators and aggregation rules.

Table 2: Performance of BigBang-V1 with and without inference-time search and verification on long-horizon search and scientific research benchmarks.

| Model | *Long-horizon Search* | | *Scientific Research* | | | |
|---|---|---|---|---|---|---|
| | **BrowseComp** | **xbench** | **FS-R** | **HLE** | **BioMystery-HS** | **BioMystery-HD** |
| **BigBang-V1** | 76.5 | 58.4 | 46.2 | 50.3 | 57.5 | 15.7 |
| **BigBang-V1 + Search-and-Verify** | **76.8** | **64.6** | **46.9** | **52.5** | **63.0** | **17.6** |

## 3.2 Quantitative Results

**State-of-the-art performance at the 35B scale.** Within the 35B model group in Table 1, BigBang-V1 obtains the highest reported score on nine of the eleven benchmarks: BrowseComp, SWE-Bench Pro, SciCode-V-Sub, FS-R, HLE, BioMystery-HS, BioMystery-HD, MLE-Bench(Lite), and PaperBench(Code-Dev). It additionally ties Agents-A1 for the best SciCode-V-Main score, placing first or joint first on nine rows across long-horizon search, coding, scientific research, and AI research. Several margins are substantial: BigBang-V1 leads the best-performing alternative 35B model by 4.0 points on SWE-Bench Pro, 7.8 points on FS-R, 7.3 points on BioMystery-HS, and 9.8 points on BioMystery-HD. On BrowseComp, it exceeds Nex-N2-mini by 2.4 points. With 35B total parameters and only 3B activated during inference, these results place BigBang-V1 at the performance frontier of its scale class across markedly different agentic tasks.

**Reaching and at times surpassing the DeepSeek V4 tier.** Against frontier reference models, BigBang-V1 outperforms DeepSeek V4 Flash Preview on six of the eleven benchmarks. It also outperforms DeepSeek V4 Pro Preview on FS-R, HLE, BioMystery-HD, and PaperBench(Code-Dev). The clearest cross-scale result is FS-R: BigBang-V1 reaches 46.2, ahead of Claude Opus 4.8 (45.2), DeepSeek V4 Pro Preview (40.7), DeepSeek V4 Flash Preview (37.7), and Step-3.7 Flash (37.2). On HLE, it scores 50.3, exceeding DeepSeek V4 Pro Preview at 48.2 and Flash Preview at 45.1; on BioMystery-HD, it reaches 15.7 compared with 13.7 for Pro Preview and 11.8 for Step-3.7 Flash. Outside scientific research, BigBang-V1 exceeds DeepSeek V4 Flash Preview on BrowseComp (76.5 versus 73.2) and SWE-Bench Pro (54.2 versus 52.6), while matching Gemini 3.1 Pro at 54.2 on SWE-Bench Pro. These comparisons place BigBang-V1 in the frontier performance range on several demanding evaluations, while the results on SciCode-Verified, BioMysteryBench, and the two AI-research benchmarks leave a clear gap to the strongest frontier systems.

**Broad gains over the Qwen3.6 backbone.** As shown in Table 1, BigBang-V1 improves over Qwen3.6-35B-A3B across all benchmarks. The largest gains are observed on FS-R, from 11.9 to 46.2 ($+34.3$); xbench, from 32.6 to 58.4 ($+25.8$); and SciCode-V-Main, from 26.6 to 50.0 ($+23.4$). The improvement also reaches 14.1 points on HLE (36.2 to 50.3), 13.7 points on BioMystery-HD (2.0 to 15.7), 12.7 points on BioMystery-HS (44.8 to 57.5), and 12.1 points on SciCode-V-Sub (56.5 to 68.6). Beyond these scientific and coding tasks, BrowseComp increases from 67.9 to 76.5, SWE-Bench Pro from 43.6 to 54.2, MLE-Bench(Lite) from 31.8 to 59.1, and PaperBench(Code-Dev) from 30.7 to 53.6.

**Transfer from verifiable frontier training.** The largest backbone gains occur on evaluations that require verifiable reasoning and sustained tool use: open-ended scientific problem solving on FS-R, long-horizon information seeking on xbench, end-to-end scientific programming on SciCode-V-Main, and difficult bioinformatics analysis on BioMystery-HD. The improvements on BrowseComp, SWE-Bench Pro, and MLE-Bench further show that the gains extend to web search, repository-level software engineering, and machine-learning engineering beyond the scientific tasks targeted directly by the training pipeline. The trajectory studies that follow examine the corresponding behavior at the task level, including exact taxon and breakpoint identification in BioMysteryBench. Table 1 measures the complete BigBang post-training

pipeline; it establishes the end-to-end gain, while the individual contributions of the generator, critic, and outcome-calibration stages require separate ablations.

**Complementary search and verification at inference time.** To examine whether inference-time search can further exploit BigBang's training on verifiable frontier tasks, the *Search-and-Verify* setting in Table 2 applies a three-way fan-out/fan-in procedure. It preserves the canonical trajectory and adds two complementary search paths: one emphasizes broad recall, while the other adversarially tests the discriminating constraints and searches for counter-evidence. Each terminal trajectory is compressed into a structured context summary that preserves source-backed evidence, candidate answers, conflicts, and unresolved questions. The reducer reconciles these summaries, and a separate no-tools finalizer produces the answer. This procedure changes only inference-time evidence collection and does not use gold answers or evaluator feedback for generation or selection. *Search-and-Verify* improves BigBang-V1 on all six reported benchmarks. The largest gains are on xbench, from 58.4 to 64.6 ($+6.2$), and BioMystery-HS, from 57.5 to 63.0 ($+5.5$). It also improves HLE from 50.3 to 52.5 ($+2.2$), BioMystery-HD from 15.7 to 17.6 ($+1.9$), FS-R from 46.2 to 46.9 ($+0.7$), and BrowseComp from 76.5 to 76.8 ($+0.3$). The pattern is consistent with the role of scientific research as a general curriculum: complementary search and verification help most when success depends on broad evidence collection or resolving competing hypotheses.

## 3.3 Qualitative Results

This section provides a qualitative comparison of BigBang and the corresponding DeepSeek variants across four benchmark families. Rather than considering only the final scores, we examine how the systems interpret evidence, revise their intermediate hypotheses, verify their own outputs, and select the final submission. The cases collectively cover biological data analysis, symbolic mathematics, paper-to-code replication, and machine-learning competition tasks.

**Biological Data Analysis** The BioMysteryBench cases evaluate whether an agent can convert noisy bioinformatics evidence into a precise and rubric-compliant biological conclusion. As shown in Table 3 and Table 4, we consider two representative tasks: identifying a viral species from intestinal-organoid RNA-seq reads and locating a short transposon insertion from whole-genome sequencing data. Across both cases, BigBang is more effective at consolidating sequence-level evidence, respecting taxonomic or coordinate conventions, and rejecting candidates that are inconsistent with the reference signal. DeepSeek V4 Flash, by contrast, returns plausible-looking biological answers but does not consistently validate them against the decisive evidence, leading to unstable virus classifications and incorrect genomic coordinates.

**Symbolic Mathematical Reasoning** The HorizonMath case examines the difference between identifying a correct special value and constructing a complete analytic derivation from the original problem. BigBang retrieves a relevant public proof strategy, verifies the nontrivial transformation identities and boundary arguments, and converts the result into the Gamma-only form required by the task. DeepSeek V4 Pro Preview reaches the same special value through exceptionally strong high-precision numerical evidence, but it does not establish the analytic bridge from the original elliptic integral to that value. BigBang's advantage is therefore not the independent discovery of a different identity, but the production of a more complete, literature-grounded, and auditable proof.

**Scientific Code Reproduction** The two PaperBench cases below isolate a mechanism that the other benchmarks cannot expose. In the Code-Dev variant used here, an agent turns a research paper into a repository that reproduces it, and a judge scores the submitted code against a hierarchical rubric without ever executing it — so nothing external forces an agent to discover that its own implementation is broken. Tables 6 and 7 show the two systems resolving that freedom differently, under the same judge (`gpt-5.5`) and the same rubric tree. BigBang runs its own pipeline and tests properties that can only hold if the method is right: it reworks the LBCS perturbation scheme after observing that the selected coreset size never falls below $k$, and it asserts the sign of the NCE gradient rather than merely that the loss evaluates. DeepSeek

Table 3: Biological data-analysis comparison on BioMysteryBench task hb031. The highlighted BigBang cells mark the evidence interpretation and answer-selection decisions most directly associated with correct viral identification.

| Analysis stage | BigBang-V1 | DeepSeek V4 Flash (max) |
|---|---|---|
| **Task:** viral species identification from three intestinal-organoid RNA-seq FASTQ files | | |
| **1. Input and task framing** | The task supplies three reduced FASTQ files from human intestinal organoid samples infected with an RNA virus. The required output is the viral species, with an all-or-nothing rubric accepting *Norovirus GII.4* or the broader label *Norovirus*. | Produced a virus label in each run, but the three answers were biologically unrelated to the accepted Norovirus answer. |
| **2. Evidence extraction** | Converged on the Norovirus family and retained the answer at the level accepted by the rubric. | Returned Severe fever with thrombocytopenia syndrome virus, Rotavirus A, and Human endogenous retrovirus K113 across the three runs. |
| **3. Candidate interpretation** | **Selected the taxon supported by the benchmark target:** Norovirus GII.4/Norovirus. The answer is accepted without requiring the subtype string. | The outputs indicate failure to validate the candidate taxon against the viral sequence signal; none of the proposed labels belongs to the accepted answer class. |
| **4. Cross-check and falsification** | The accepted answer was reproduced in all three runs, yielding stable species-level identification. | The answer changed across runs and remained incorrect in every run, indicating both low accuracy and low taxonomic stability. |
| **5. Final biological call** | Alternative virus labels were not retained because they were inconsistent with the reference answer and rubric. | The three alternatives were not near misses within the Norovirus genus; they represent substantive classification errors across distinct viral groups. |
| **6. Rubric-level verification** | **Final answer:** Norovirus GII.4/Norovirus. **Rubric status: accepted in all three runs.** | Final answers: Severe fever with thrombocytopenia syndrome virus; Rotavirus A; Human endogenous retrovirus K113. **Rubric status: rejected in all three runs.** |
| **7. Benchmark outcome** | **BioMysteryBench: 3/3 correct.** | **BioMysteryBench: 0/3 correct.** |

V4 Flash builds a comparably complete file tree but checks only that the code imports and compiles, and when an assertion fails it weakens the assertion rather than the code. What separates the two is thus the standard each holds its own work to, not how much of the paper it read.

**Machine Learning Model Development** To understand what drives BigBang's gains beyond the aggregate score, we compare its trajectories with DeepSeek V4 Flash on two tasks for which both agents produced valid submissions. Tables 8 and 9 trace how each agent moved from data inspection to its selected submission. Throughout the tables, *validation set* denotes an agent's local development split, *test set* denotes the held-out MLE-Bench evaluation, and *human leaderboard* denotes the rank among human competition entries. Because the two agents use different local validation protocols, their validation scores explain decisions within a trajectory but should not be read as a controlled head-to-head result.

# 4 Conclusion

In this report, we introduce a self-evolving data synthesis framework for training models on verifiable frontier tasks. Using data generated by this framework, BigBang-V1 substantially improves over its base model across scientific research, reasoning, coding, search, and tool use. It outperforms DeepSeek V4

Table 4: Biological data-analysis comparison on BioMysteryBench task rechlnhmn1jsmsser. The highlighted BigBang cells mark the breakpoint-localization and coordinate-selection decisions most directly associated with correct insertion-site identification.

| Analysis stage | BigBang-V1 | DeepSeek V4 Flash (max) |
|---|---|---|
| **Task:** locate the 47-bp transposon insertion in Klebsiella pneumoniae WGS reads | | |
| **1. Input and task framing** | The task provides two WGS FASTQ files and the reference sequence `kpneumoniae_ref.fasta`. The output must contain the RefSeq chromosome and a 1-based coordinate. | Returned a genomic coordinate in each run, but did not recover the reference-supported insertion breakpoint. |
| **2. Evidence extraction** | The successful runs resolved the transposon–chromosome junction and identified chromosome `NC_009648.1`. | Reported positions 1, 1,590,256, and 4,660,944 across the three runs. |
| **3. Candidate interpretation** | **Changed from a generic coordinate guess to junction-constrained localization:** the accepted breakpoint is `NC_009648.1:4,144,431`, using the requested 1-based convention. | The reported positions did not stabilize on the insertion junction. The errors are not explained by a 0-based versus 1-based conversion. |
| **4. Cross-check and falsification** | Recovered the exact chromosome-position pair in two of three runs, showing substantial but imperfect reproducibility for precise breakpoint analysis. | All three runs missed the required coordinate, with one boundary-like value and two distant genomic positions. |
| **5. Final biological call** | Candidate coordinates that did not match the transposon junction and the reference accession were rejected in the successful runs. | None of the three candidate coordinates matched the gold breakpoint; the discrepancy is a localization failure rather than a formatting issue. |
| **6. Rubric-level verification** | **Final answer:** chromosome `NC_009648.1`, position 4,144,431, corresponding to the 47-bp insertion. **Accepted in two of three runs.** | Final positions: 1; 1,590,256; and 4,660,944. **Rejected in all three runs.** |
| **7. Benchmark outcome** | **BioMysteryBench: 2/3 correct.** | **BioMysteryBench: 0/3 correct.** |

Flash Preview overall and sets a new state of the art among models at the 35B scale. On several tasks, it even surpasses DeepSeek V4 Pro Preview, which has more than one trillion parameters.

These results suggest that sustained capability growth depends not only on model scale or data quantity, but also on the ability of the training process to continually discover challenging, diverse, and objectively verifiable tasks. By treating autonomous scientific research as both a target capability and a source of open-ended training signals, BigBang represents an early step toward scalable AI4AI and grounded recursive self-improvement.

**What's next.** Our internal experiments suggest that the current self-evolving pipeline is broadly applicable and has strong scaling potential. Moving forward, we will continue to refine and scale this pipeline to synthesize more diverse and challenging training data, with the goal of further advancing the capability frontier of general-purpose models.

Table 5: Trajectory comparison on **Spinor Norm Elliptic Integral**. Both systems return a correct closed form. The difference is that BigBang-V1 turns a retrieved public proof outline into an auditable symbolic derivation and checks its intermediate identities, whereas DeepSeek V4 Pro Preview establishes a highly accurate numerical match but does not derive the bridge from the original integral to the special value.

| **Trajectory stage** | **BigBang-V1** | **DeepSeek V4 Pro Preview (max)** |
|---|---|---|
| **Task:** evaluate $I_0 = \frac{4\sqrt{2}}{\pi}\int_0^1 \frac{1+\sqrt{z}}{(1+z)^3}\big[2E(z)-(1-z)K(z)\big]dz$ without returning a numerical integral or a truncated series. | | |
| **1. Initial evidence** | **Finds the decisive public proof source.** Searches the exact integrand, `spinor norm elliptic integral`, and $I_0 \approx 1.77425$. After one dead OpenReview link, it finds the appendix whose extracted text includes the target value, generated code, and a derivation outline. | Searches the original Takahashi spinor-norm paper, the numerical value, and related elliptic-integral queries. The primary source gives only $I_0 \simeq 1.774$ and explicitly says that the closed form is unknown; no closed-form proof is retrieved. |
| **2. What the evidence contains** | The retrieved appendix states both $I_0 = 2E(1/2) - K(1/2)/2$ and the Gamma expression. It also supplies the intended proof ingredients: Landen, a derivative identity, a Beltrami/imaginary-modulus transform, and the $m = 1/2$ singular values. | Establishes an 80-digit numerical target by quadrature. It tests simple combinations of $\pi$, Catalan's constant, logarithms, and Gamma values; polynomial searches through degrees 2–7 and a PSLQ-style basis search return no relation. |
| **3. Landen step** | **Checks the relevant convention, rather than merely quoting it:** $2E(t^2)-(1-t^2)K(t^2) = (1+t)E(4t/(1+t)^2)$. With $z = t^2$, it uses this to replace the mixed $K, E$ integrand by a single transformed $E$-integrand. | Looks up Landen transformations and tests the ascending modulus $2\sqrt{k}/(1+k)$ and descending complementary modulus. The trace explicitly encounters factor-of-two and parameter/modulus mismatches, but does not settle on a transformation that reduces the original integral. |
| **4. Reduction to a $K$-moment** | **Converts the target integral, with a checked boundary argument.** Sets $A(x) = E(1-x^2)$, $B(x) = K(1-x^2)$, and verifies the corrected identity $\frac{d}{dx}\frac{x(A-B)}{(1+x^2)^2} = \frac{2(1-x^2)A+(3x^2-1)B}{(1+x^2)^3}$. Integrating the vanishing boundary term converts the target $E$-moment to a rationally weighted $K$-moment. | Uses the rational substitution $\sqrt{z} = (1-u)/(1+u)$ and computes its Jacobian symbolically. This produces a transformed weight, but the elliptic factor is not reduced. The trajectory stops before an integration-by-parts identity or a boundary-term argument is obtained. |
| **5. Parameter transform** | **Introduces a parameter family that closes the proof.** Defines $F(k) = \int_0^1 \frac{kK(1-x^2)}{k^2+x^2}\,dx$, verifies $I_0 = 2\sqrt{2}F''(1)/\pi$, and checks $F(k) = \frac{\pi}{2\sqrt{1+k^2}}K\Big(\frac{1}{1+k^2}\Big)$. Differentiating gives $I_0 = 2E(1/2) - K(1/2)/2$. | Directly tests $2E(1/2) - K(1/2)/2$ against quadrature. It obtains agreement at about 80 digits and later about 200 digits. This is excellent numerical evidence, but no $F(k)$-type transform, differentiation argument, or analytic equality is supplied. |
| **6. Special-value closure** | **Closes in the permitted symbolic language.** Evaluates $K(1/2) = \Gamma(1/4)^2/(4\sqrt{\pi})$, applies the self-complementary Legendre relation for $E(1/2)$, and uses $\Gamma(1/4)\Gamma(3/4) = \pi\sqrt{2}$ to obtain a Gamma-only result. | Checks $K(1/2)$'s Gamma special value, tries several tentative formulas for $E(1/2)$, and eventually verifies an equivalent Gamma rearrangement numerically. Its submitted program nevertheless retains `ellipk(1/2)` and `ellipe(1/2)`, rather than documenting the singular-value derivation. |
| **7. Final deliverable** | **Returns** $\frac{\Gamma(1/4)^2}{8\sqrt{\pi}} + \frac{\Gamma(3/4)^2}{\sqrt{\pi}}$. **The answer is Gamma-only and accompanied by a checkable proof-and-verification chain.** | **Returns** $2E(1/2) - \frac{1}{2}K(1/2)$. **The value is correct and strongly validated numerically, but the trajectory contains no complete analytic proof from the original integral.** |

Table 6: Detailed trajectory comparison on **Refined Coreset Selection** (LBCS). The highlighted BigBang cells mark the self-correction and consistency-checking decisions that most directly preserved the highest-weight rubric leaf.

| Trajectory stage | BigBang-V1 | DeepSeek V4 Flash (max) |
|---|---|---|
| **Task:** implement lexicographic bi-level coreset selection under a performance constraint | | |
| **1. Paper ingestion** | Read `blacklist.txt` and `addendum.md` before the paper itself, confirmed the official repository was off limits, then extracted the PDF and read all 2,301 lines of text in five ordered passes. | Read the same preliminaries, then extracted the PDF and consumed the whole 80 KB body in a single `cat`. Planning was correct in substance: it identified the $f_1/f_2$ lexicographic objective and the out-of-scope appendix sections. |
| **2. Decomposition** | Recorded a structured summary in `notes/paper_notes.md` and fixed a three-layer layout that keeps Algorithm 2 (LexiFlow) a separate module invoked from step 4 of Algorithm 1 — the boundary the rubric's heaviest leaf tests. | Wrote 30 files in one uninterrupted pass, laying out an equivalent module set. `mask_optimizer.py` implements LexiFlow in 250 lines and is exported from the package. |
| **3. Self-diagnosis** | **Detected a violation of the paper's central claim:** smoke tests showed $f_2$ pinned at $k = 100$, so the coreset could never shrink. It derived that the perturbation scale $\delta u_i \approx 0.0004$ could not cross the $0.5$ threshold, and rejected three candidate fixes in turn. | Ran `py_compile` and an import check, then proceeded to packaging. No execution of the pipeline was attempted, so the equivalent defect could not surface. |
| **4. Repair** | **Replaced continuous perturbation with binary mask flipping** and confirmed the selected size now moved (121, 102, 154; end-to-end 200 $\rightarrow$ 188). A second pass found a REINFORCE term detached from the graph and restored gradient flow. | Rewrote `core.py` once to track Algorithms 1–2 more closely. The rewrite dropped the call into the LexiFlow module, leaving its 250 lines unreachable — the single defect that forfeits the heaviest leaf (weight $0.163$). |
| **5. Consistency sweep** | One baseline exposed a positional-argument mismatch in `Mask(n)`. Rather than patch in place, it grepped the repository, fixed five instances of the same error, and re-checked the call sites in the experiment scripts. | Never invoked `experiments/`. Three scripts pass `train_dataset=` to a constructor expecting `dataset=`, and unpack three values from a five-value return; every Section 5.1 leaf therefore fails. |
| **6. Final selection** | Submitted after 102 turns and 43.9 minutes, documenting the perturbation deviation in the README. The judge's comment records "minor deviations noted but required structure implemented". **331 of 485 leaves pass.** | Submitted after 66 turns and 9.1 minutes, using 22% of its step budget. Passing leaves are concentrated in paper-independent scaffolding — dataset loaders, model definitions, a standalone Figure 1 script. **170 of 485 leaves pass.** |
| **7. Final result** | **Replication score:** $0.7657$. **Algorithm 1+2 leaf (weight $0.163$): full credit.** | **Replication score:** $0.3053$. Algorithm 1+2 leaf (weight $0.163$): zero credit. |

Table 7: Detailed trajectory comparison on **BBox-Adapter**. The decisive difference is whether verification tests behaviour or merely confirms that the code imports.

| Trajectory stage | BigBang-V1 | DeepSeek V4 Flash |
|---|---|---|
| **Task:** adapt a black-box LLM with an energy-based scorer trained by ranking NCE | | |
| **1. Constraint reading** | Opened `addendum.md` before the paper and acted on its clarification that the spectral normalisation of Eq. 3 is an $L_2$ penalty on the energy rather than power iteration, implementing it that way. | Read the addendum in the same opening block and also adopted the $L_2$ reading; the NCE and normalisation leaves are among the few it passes. |
| **2. Prioritisation** | Transcribed the eight prompts of Appendix J before writing any method code, judging them zero-inference, directly checkable rubric items. All three chain-of-thought baseline leaves pass. | Wrote 15 files in a single uninterrupted pass with no planning turn. Across 65 model calls it emitted 3,110 reasoning tokens and no plan, rubric self-check, or coverage review. |
| **3. Verification** | **Asserted behaviour, not executability:** it fed pos $= [2, 3]$, neg $= [-1, -0.5]$ through the loss and required the positive-sample gradient to be negative and the negative-sample gradient positive, i.e. that Eq. 3 truly raises positive energy. The whole NCE subtree passes 5/5. | An assertion `loss.item() > 0` failed. It rewrote the test file to delete the assertion, printed `loss = -3.0667`, and declared the check passed. |
| **4. Defect tracing** | A self-test raised a rank mismatch ("got 3 and 2"). It traced the cause to single-sample forwards returning $[1]$, stacking to $[2, 1]$ and then being unsqueezed, patched both call sites, and re-ran to confirm the loss became finite and differentiable. | Every LLM call routes through a placeholder callback that logs a warning; no OpenAI or HuggingFace client was ever written. Eight further leaves covering decoding temperature, context length and the three model backends fail as a consequence. |
| **5. Specification audit** | Rechecked the paper's metric definitions late in the run, found that TruthfulQA is scored by True+Info rather than exact match, and added the corresponding two-dimensional judge. | Set `training_steps_per_iteration=1000` against the paper's 6000, then divided again by batch size inside the loop, yielding 15 optimiser steps; this single error forfeits all five backbone configuration leaves. |
| **6. Final selection** | Stopped chasing two dataset downloads that failed for environmental reasons and spent the remaining turns on `requirements`, `reproduce.sh` and the README — the correct trade under a judge that reads but never runs the code. **Submitted after 91 turns; 70 of 145 leaves pass.** | Submitted after 65 turns and 8.2 minutes, using 22% of its step budget. All five experiment tables, jointly worth 0.538 of the paper's weight, are absent. **33 of 145 leaves pass.** |
| **7. Final result** | **Replication score:** 0.5402. **Algorithm 1 subtree:** 0.767. | **Replication score:** 0.2132. Algorithm 1 subtree: 0.572; all experiment tables zero. |

Table 8: Detailed trajectory comparison on **MLSP 2013 Birds**. The highlighted BigBang cells mark the representation change and selection decisions that most directly improved generalization.

| Trajectory stage | BigBang-V1 | DeepSeek V4 Flash |
|---|---|---|
| **Task:** multi-label bird-call recognition from audio | | |
| **1. Data diagnosis** | Identified a small, imbalanced problem with 258 labeled recordings and 19 species. Inspected raw audio together with provided mel, filtered-spectrogram, histogram, and segment representations. | Prioritized the supplied 100-D histogram and 38-D segment features, noting that segment features were unavailable for many recordings and that several species had very few positives. |
| **2. Baselines** | A histogram–LightGBM baseline reached only 0.6093 mean AUC on the **validation set**, indicating that compact summary features discarded important spectro-temporal structure. | Logistic regression reached 0.7566, while tuned XGBoost and random forests each reached about 0.7682 mean AUC on the **validation set**. |
| **3. Model pivot** | **Changed the representation:** trained ResNet18 directly on mel spectrograms with masking, noise, amplitude scaling, and mixup. A single 128-bin mel CNN reached 0.8417 mean AUC on the **validation set**. | Retained the hand-crafted representation. Averaging XGBoost and random forest predictions reached 0.7758; adding segment-level MIML XGBoost with max pooling raised AUC on the validation set to 0.7959. |
| **4. Robustness** | Averaged three seeds of the main mel CNN, raising AUC on the validation set to 0.8615, and trained complementary filtered-spectrogram and high-resolution mel models to diversify errors. | Combined five-seed XGBoost, random forest, and MIML predictions, reaching 0.8081 AUC on the validation set. Adding more boosting seeds and a weaker gradient-boosting model reduced it to 0.7963. |
| **5. Falsification** | Tested DenseNet121 (0.7763), EfficientNet-B0 (0.7202), focal loss (0.8295), class-weighted BCE (0.8104), and alternate mel parameters (0.8068) on the validation set; all were rejected. | Tested mean rather than max MIML pooling (0.8031), classifier chains (0.7064), and label co-occurrence smoothing (0.8006) on the validation set; none surpassed the selected ensemble. |
| **6. Final selection** | Fine-grained weight search combined the three-seed mel, two-seed filtered-spectrogram, high-resolution mel, and provided-mel CNNs with weights $(0.48, 0.40, 0.08, 0.04)$. **Validation set:** 0.8736 **mean AUC.** | Selected the average of multi-seed XGBoost, random forest, and MIML max-pooling predictions, with no end-to-end spectrogram learner in the final pipeline. **Validation set:** 0.8081 **mean AUC.** |
| **7. Final result** | **Test set (MLE-Bench):** 0.90708 **AUC.** **Human leaderboard rank: 15/81; silver medal.** | **Test set (MLE-Bench):** 0.85641 **AUC.** Human leaderboard rank: 45/81; no medal. |

Table 9: Detailed trajectory comparison on **Russian Text Normalization**. The decisive difference is whether validation exposes and drives improvement on surface forms absent from the lookup table.

| Trajectory stage | BigBang-V1 | DeepSeek V4 Flash (max) |
|---|---|---|
| **Task:** token-level normalization of numbers, dates, units, and transliterated text | | |
| **1. Validation design** | Built the frequent-token lookup on a 95% training split and evaluated on the remaining 5%. It separately tracked accuracy on surface forms absent from the lookup and bucketed transliteration, date, phone, and other errors. | Built the `before`→most-common-`after` table from all 9.5M training rows, then sampled rows from that same data for its diagnostic. Most sampled tokens were therefore already present in the lookup. |
| **2. Lookup baseline** | Most-common lookup with identity fallback achieved 0.973348 **accuracy on the validation set**; the explicit unseen slice exposed the remaining generalization gap. | The full-data lookup reported about 0.9902 on its in-sample diagnostic. Unknown test tokens were simply returned unchanged, and no held-out unseen-token diagnostic was used for selection. |
| **3. Model search** | Tested identity-only (87.53%), a sequence model (92.86%), and overly broad rules (96.57%) on the validation set; their regressions motivated a hybrid of precise lookup and targeted rules. | Tested classifier-conditioned lookup (about 93.9%), neural character models (up to about 99.016%), and context memories. Their marginal gains or regressions led back to the global lookup. |
| **4. Unseen transliteration** | **Optimized the unseen tail:** learned a 105K-entry word transliteration lexicon plus context-conditioned character mappings and ordered digraph rules. Accuracy on the validation set rose from 0.980743 to 0.980862, while unseen accuracy rose from 81.3% to 82.5%. | Kept the identity fallback for unknown tokens. Character and context alternatives were judged mainly by aggregate accuracy, so errors on rare unseen forms did not become a sustained optimization target. |
| **5. Morphology & format** | Corrected ordinal case, year abbreviations, feminine number forms, decimals, units, currencies, dates, mixed scripts, and space-separated numbers. These targeted repairs raised accuracy on the validation set to 0.981031. | Rejected context-based disambiguation as too sparse and neural normalization as expensive for marginal aggregate gains; it did not build a structured Russian normalizer for unseen tokens. |
| **6. Final selection** | Used residual error buckets to repair ordered digraphs, ordinal suffixes, magnitude forms, and leading-zero phone groups. Lookup plus rules finished at 0.981373 **accuracy on the validation set**, with about 83.5% unseen accuracy. | Returned to the single global lookup after the alternatives failed to improve its in-sample diagnostic. **Local in-sample diagnostic: approximately** 0.9902 **accuracy**; this is not a held-out validation score. |
| **7. Final result** | **Test set (MLE-Bench):** 0.98083 **accuracy. Human leaderboard rank: 37/163; bronze medal.** | **Test set (MLE-Bench):** 0.97348 **accuracy.** Human leaderboard rank: 122/163; no medal. |

Across both tasks, BigBang's advantage comes from changing the problem representation in response to diagnostic evidence rather than only tuning the current pipeline. On Birds, it recognizes that aggregate hand-crafted audio features are the bottleneck, moves to end-to-end spectrogram learning, and then reduces variance with multi-seed, multi-view weighted averaging; this produces a 0.05067 held-out AUC advantage. On Russian normalization, it notices that high aggregate lookup accuracy masks poor behavior on unseen forms, explicitly tracks that slice, and turns the residual errors into targeted transliteration and morphology rules; DeepSeek V4 Flash instead selects an in-sample-validated lookup whose identity fallback does not generalize to the unseen tail. The two cases therefore expose the same transferable strength: BigBang uses validation failures to identify the right abstraction to change, then preserves complementary solutions through disciplined selection and ensembling, yielding both a silver and a bronze result where DeepSeek V4 Flash earns no medal.

## References


Anthropic. Introducing claude opus 4.8, May 2026a. URL https://www.anthropic.com/news/claude-opus-4-8.

Anthropic. Evaluating claude's bioinformatics research capabilities with biomysterybench, April 2026b. URL https://www.anthropic.com/research/Evaluating-Claude-For-Bioinformatics-With-BioMysteryBench.

Apodex Team. Apodex-1.0: A verification-centric agent team for discoverative intelligence, June 2026. URL https://www.apodex.com/blog/apodex-1.0.

Lei Bai, Zongsheng Cao, Yang Chen, Zhiyao Cui, Shangheng Du, Yue Fan, Shiyang Feng, Zijie Guo, Haonan He, Liang He, Xiaohan He, Shuyue Hu, Yusong Hu, Songtao Huang, Yichen Jiang, Hao Li, Xin Li, Dahua Lin, Weihao Lin, Fenghua Ling, Dongrui Liu, Zhuo Liu, et al. Scaling the horizon, not the parameters: Reaching trillion-parameter performance with a 35b agent. *arXiv preprint arXiv:2606.30616*, 2026. URL https://arxiv.org/abs/2606.30616.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Lilian Weng, and Aleksander Mądry. Mle-bench: Evaluating machine learning agents on machine learning engineering. *arXiv preprint arXiv:2410.07095*, 2025. URL https://arxiv.org/abs/2410.07095.

Leonardo de Moura and Sebastian Ullrich. The Lean 4 theorem prover and programming language. In André Platzer and Geoff Sutcliffe, editors, *Automated Deduction – CADE 28*, volume 12699 of *Lecture Notes in Computer Science*, pages 625–635, Cham, 2021. Springer. doi: 10.1007/978-3-030-79876-5_37.

DeepSeek Team. Deepseek v4 preview release, April 2026. URL https://api-docs.deepseek.com/news/news260424/.

Xiang Deng, Jeff Da, Edwin Pan, Yannis Yiming He, Charles Ide, Kanak Garg, Niklas Lauffer, Andrew Park, Nitin Pasari, Chetan Rane, Karmini Sampath, Maya Krishnan, Srivatsa Kundurthy, Sean Hendryx, Zifan Wang, Vijay Bharadwaj, Jeff Holm, Raja Aluri, Chen Bo Calvin Zhang, Noah Jacobson, Bing Liu, and Brad Kenstler. Swe-bench pro: Can ai agents solve long-horizon software engineering tasks? *arXiv preprint arXiv:2509.16941*, 2025. URL https://arxiv.org/abs/2509.16941.

Yaxin Du, Xiyuan Yang, Zhifan Zhou, Wanxu Liu, Zixing Lei, Zimeng Chen, Fenyi Liu, Haotian Wu, Yuzhu Cai, Zexi Liu, et al. Datamaster: Data-centric autonomous ai research. *arXiv preprint arXiv:2605.10906*, 2026a.

Yuwen Du, Rui Ye, Shuo Tang, Keduan Huang, Xinyu Zhu, Yuzhu Cai, and Siheng Chen. Openseeker-v2: Pushing the limits of search agents with informative and high-difficulty trajectories. *arXiv preprint arXiv:2605.04036*, 2026b.

Yuwen Du, Rui Ye, Shuo Tang, Xinyu Zhu, Yijun Lu, Yuzhu Cai, and Siheng Chen. Openseeker: Democratizing frontier search agents by fully open-sourcing training data. *arXiv preprint arXiv:2603.15594*, 2026c.

Google DeepMind. Gemini 3.1 pro model card, February 2026. URL https://deepmind.google/models/model-cards/gemini-3-1-pro.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, et al. Accelerating scientific discovery with Co-Scientist. *Nature*, 655:487–496, 2026. doi: 10.1038/s41586-026-10644-y. URL https://doi.org/10.1038/s41586-026-10644-y.

Sihan Hu, Lyuhan Huang, Youjin Deng, and Kun Chen. SciCode-Verified: How benchmark defects underestimated the scientific-coding ability of language models. *arXiv preprint arXiv:2608.04975*, 2026. doi: 10.48550/arXiv.2608.04975. URL https://arxiv.org/abs/2608.04975.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021. doi: 10.1038/s41586-021-03819-2.

Nestor Maslej, Loredana Fattorini, Raymond Perrault, Yolanda Gil, Vanessa Parli, Njenga Kariuki, Emily Capstick, Anka Reuel, Erik Brynjolfsson, John Etchemendy, Katrina Ligett, Terah Lyons, James Manyika, Juan Carlos Niebles, Yoav Shoham, Russell Wald, Tobi Walsh, Armin Hamrah, Lapo Santarlasci, Julia Betts Lotufo, Alexandra Rome, Andrew Shi, and Sukrut Oak. Artificial intelligence index report 2025. Technical report, Stanford Institute for Human-Centered Artificial Intelligence, 2025. URL https://arxiv.org/abs/2504.07139.

Nex AGI. Nex-n2: Thinking, built for action, June 2026. URL https://nex-agi.com/.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. AlphaEvolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025. doi: 10.48550/arXiv.2506.13131. URL https://arxiv.org/abs/2506.13131.

OpenAI. GPT-5.5 system card, April 2026. URL https://deploymentsafety.openai.com/gpt-5-5/gpt-5-5.pdf.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, Michael Choi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025a. doi: 10.48550/arXiv.2501.14249. URL https://arxiv.org/abs/2501.14249.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025b.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all, April 2026. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b.

Christopher J. Roy and William L. Oberkampf. A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing. *Computer Methods in Applied Mechanics and Engineering*, 200(25–28):2131–2144, 2011. doi: 10.1016/j.cma.2011.03.016.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. Paperbench: Evaluating ai's ability to replicate ai research. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 56843–56873. PMLR, 2025. URL https://proceedings.mlr.press/v267/starace25a.html.

StepFun. Step 3.7 flash: A high-efficiency flash model for real-world agents, May 2026. URL https://static.stepfun.com/blog/step-3.7-flash/.

Minyang Tian, Luyu Gao, Shizhuo Dylan Zhang, Xinan Chen, Cunwei Fan, Xuefei Guo, Roland Haas, Pan Ji, Kittithat Krongchon, Yao Li, Shengyan Liu, Di Luo, Yutao Ma, Hao Tong, Kha Trinh, Chenyu Tian, Zihan Wang, Bohao Wu, Yanyu Xiong, Shengzhu Yin, Minhui Zhu, Kilian Lieret, Yanxin Lu, Genglin Liu, Yufeng Du, Tianhua Tao, Ofir Press, Jamie Callan, Eliu Huerta, and Hao Peng. Scicode: A research coding benchmark curated by scientists. *arXiv preprint arXiv:2407.13168*, 2024. URL https://arxiv.org/abs/2407.13168.

Miles Wang, Robi Lin, Kat Hu, Joy Jiao, Neil Chowdhury, Ethan Chang, and Tejal Patwardhan. Frontierscience: Evaluating ai's ability to perform expert-level scientific tasks. *arXiv preprint arXiv:2601.21165*, 2026. URL https://arxiv.org/abs/2601.21165.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025.

Xbench-Team. Xbench-deepsearch, 2025. URL https://xbench.org/agi/aisearch.

John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik R Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.15793.

Z.ai. GLM-5.2: Built for long-horizon tasks, June 2026. URL https://z.ai/blog/glm-5.2.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data. *arXiv preprint arXiv:2505.03335*, 2025. doi: 10.48550/arXiv.2505.03335. URL https://arxiv.org/abs/2505.03335.

# A Contributions and Acknowledgments

## A.1 Projects

*Names are listed alphabetically by surname within each project.*

**Long-horizon Search**: Yuwen Du, Rui Ye

**Agentic Coding**: Guo Chen, Yifeng Gao, Xiangqi Jin, Jiacheng Liu, Junxi Wang, Ziyue Wang, Shaoqiu Zhang, Jiayi Zhu

**Scientific Research**: Jingyi Chai, Xiangrui Liu, Xianghe Pang, Yuhong Sun, Shuo Tang

**AI Research**: Yuzhu Cai, Xiyan Gui, Wenkai Jin, Fengyang Li, Zexi Liu, Cheng Wang, Xinyu Zhu

**Agent Harness**: Zexi Liu, Cheng Wang

**Evaluation and Benchmarking**: Xiangrui Liu, Cong Wang, Sihan Hu

**Co-Evolution Data Pipeline**: Xianghe Pang, Shuo Tang

**Infra**: Yuzhu Cai, Yulang Chen, Jiacheng Liu, Fenyi Liu, Jingwei Song, Shuo Tang, Yicun Yang, Xinyu Zhu

## A.2 Scientific Directors and Advisors

Kun Chen, Weinan E, Jingtao Han, Ruoxue Liao, Linfeng Zhang

## A.3 Project Co-leads

Siheng Chen, sihengc@sjtu.edu.cn

Linfeng Zhang, zhanglinfeng@sjtu.edu.cn