InteractScience: Programmatic and Visually-Grounded Evaluation of Interactive Scientific Demonstration Code Generation

Qiaosheng Chen [1 2], Yang Liu [1], Lei Li [3], Kai Chen [2], Qipeng Guo [2], Gong Cheng [1 †], Fei Yuan [2 †]

[1] State Key Laboratory for Novel Software Technology, Nanjing University
[2] Shanghai Artificial Intelligence Laboratory
[3] Carnegie Mellon University

E-mail: qschen@smail.nju.edu.cn

The Evaluation Gap

- ***Static Benchmarks***: Limited to text-only QA or static UI generation (HTML/CSS).
- ***The Integration Gap***: LLMs fail to bind user interactions with dynamic, accurate scientific principles.
- ***Superficial Competence***: Models generate visually plausible layouts that are scientifically broken.

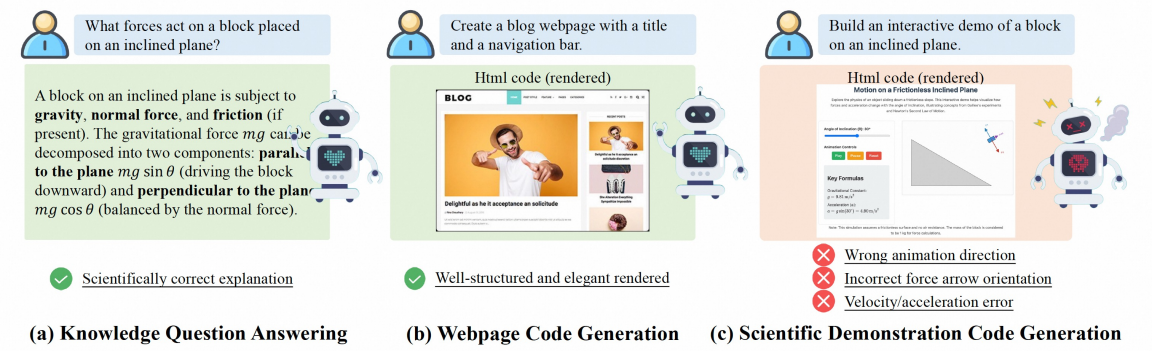


Figure 1. Illustration of three tasks. (a) **Knowledge Question Answering**: given the query about forces act of a block placed on an inclined plane, an LLM can output a correct textual explanation. (b) **Webpage Code Generation**: given the instruction of write a blog webpage, an LLM can generate functional static HTML code. (c) **Scientific Demonstration Code Generation**: generating an interactive demo for the inclined plane scenario, an LLM often fail to produce correct results.

Core Contributions of Our Work

- ***Framework***

A dual-track protocol coupling Programmatic Functional Testing (PFT) and Visually-Grounded Qualitative Testing (VQT).

- ***Benchmark***

InteractScience — the first benchmark featuring 150 diverse tasks for interactive scientific demonstration code generation.

- ***Insights***

A massive evaluation of 30 SOTA LLMs, revealing critical failure patterns in scientific reasoning and webdev coding.

Table 1. Comparison of InteractScience with related scientific visualization, software engineering, and code generation benchmarks. **SR**, **IT**, and **VT** denote Scientific Reasoning, Interaction-based Testing, and Visualization-based Testing, respectively.

Benchmark	Primary Task	SR	IT	VT	Evaluation
For Scientific Visualization:					
SridBench	Scientific Visualization Generation	✓	✗	✓	LLM-judge
EduVisBench	Scientific Visualization Generation	✓	✗	✓	LLM-judge
For Software Engineering:					
SWE-bench	Repository-level Bug Fixing	✗	✓	✗	Automation
SWE-bench Multimodal	Repository-level Bug Fixing with Visual Context	✗	✓	✗	Automation
For Code Generation:					
LiveCodeBench	Competitive-Programming Code Generation	✗	✗	✗	Automation
ChartMimic	Chart Code Generation	✗	✗	✓	Automation + LLM-judge
MatPlotBench	Chart Code Generation	✗	✗	✓	LLM-judge
PandasPlotBench	Chart Code Generation	✗	✗	✓	LLM-judge
WebDev Arena	Web Design	✗	✗	✓	Human Voting
Interaction2Code	Interactive Web Page Generation	✗	✓	✓	Automation
WebGen-Bench	Interactive Web Page Generation	✗	✗	✓	LLM-judge
ArtifactsBench	Interactive Visual Artifacts	✗	✗	✓	LLM-judge
InteractScience	Interactive Scientific Demonstration Generation	✓	✓	✓	Automation + LLM-judge

Benchmark Construction & Dataset Statistics

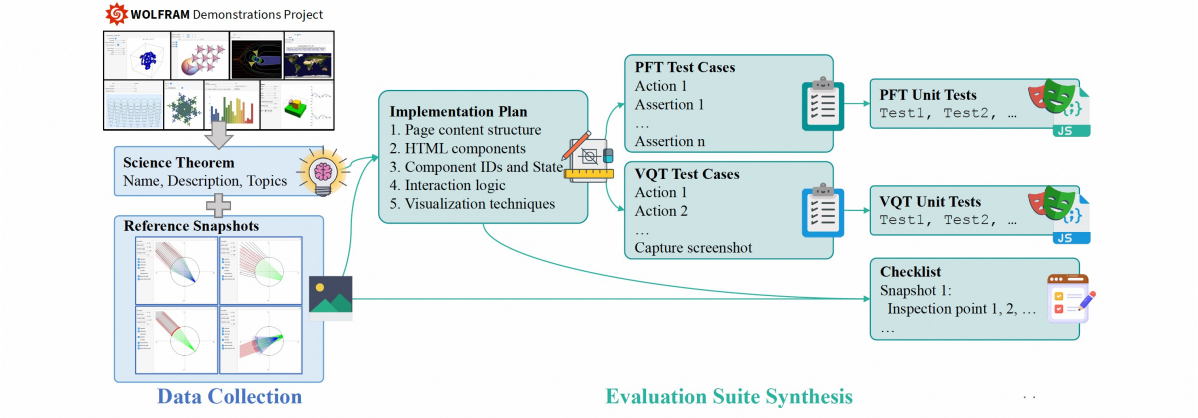


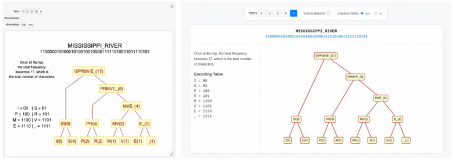
Figure 2. Pipeline of data collection and evaluation suite synthesis. The data collection step retrieves metadata of scientific demonstrations and corresponding snapshots from the Wolfram Demonstrations Project as seed data. The evaluation suite synthesis step generates implementation plans, test cases, unit tests, and checklist sequentially from the seed data.

- ***Source***: 150 complex simulations curated from the *Wolfram Demonstrations Project* across 5 major disciplines.
- ***Test Suite***:
  - Fine-grained plans to guide the generation.
  - Automated testing code anchored by checklist and reference snapshots.
- ***Granularity***: Stratified into Easy, Medium, and Hard tiers, requiring up to 15+ automated user actions per test.

Experimental Results & Analysis

Table 3. Main results of 10 closed-source and 20 open-source models on the InteractScience benchmark. CLIP and VLM-Judge scores are normalized to a 0–100 scale.

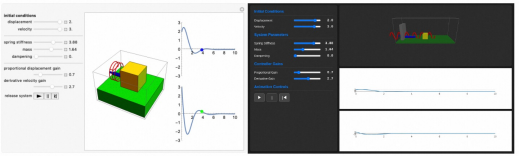
Model	PFT			VQT		
	Overall (%)	Average (%)	Perfect (%)	VLM-Judge	CLIP	Action (%)
Closed-Source Large Language Models						
GPT-5	39.47	**37.61**	**16.08**	**57.02**	71.95	89.66
GPT-4.1	37.07	34.08	11.19	52.84	71.21	89.15
GPT-4o	28.27	27.09	5.59	42.45	67.11	85.93
o3	34.93	32.09	13.99	52.82	72.24	89.83
o4-mini	37.33	34.90	13.29	51.90	71.79	88.64
Gemini-2.5-Pro	35.33	34.62	11.19	54.69	70.65	86.78
Gemini-2.5-Flash	31.60	31.07	10.49	49.34	69.59	86.95
Claude-Sonnet-4-20250514	**41.47**	37.40	13.29	55.42	73.50	89.66
Claude-Opus-4-20250514	40.27	36.34	11.19	54.93	**73.22**	89.32
Claude-3.5-Sonnet	33.33	31.45	9.79	49.43	72.32	**90.17**
Open-Source Large Language Models						
DeepSeek-R1-0528	**33.87**	**32.02**	8.39	49.46	69.54	88.31
DeepSeek-V3-0324	31.73	30.57	10.49	49.46	68.68	85.93
Kimi-K2	31.60	31.22	9.79	**50.04**	70.11	87.29
GLM-4.5	29.33	26.65	8.39	38.57	55.90	70.51
Intern-S1	31.87	28.93	7.69	45.27	68.74	87.46
gpt-oss-120b	28.00	27.78	9.79	49.57	**72.13**	**90.85**
gpt-oss-20b	15.20	12.97	3.50	21.40	54.68	80.51
Qwen3-235B-A22B-Instruct-2507	33.33	31.46	**13.29**	45.14	70.02	78.14
Qwen3-32B	27.20	24.09	5.59	39.69	66.46	87.46
Qwen3-14B	24.13	23.58	7.69	36.53	66.46	85.08
Qwen3-8B	20.00	18.85	4.20	34.67	64.13	81.53
Qwen3-4B	14.67	13.10	2.80	28.33	60.90	82.03
Qwen3-1.7B	6.53	6.22	1.40	20.33	59.65	75.76
Qwen2.5-Coder-32B-Instruct	27.20	25.10	7.69	38.51	51.67	84.58
Qwen2.5-Coder-14B-Instruct	22.53	20.61	4.90	35.72	64.47	85.42
Qwen2.5-Coder-7B-Instruct	12.40	10.51	0.70	26.97	65.17	82.37
Qwen2.5-VL-72B-Instruct	23.73	22.82	6.99	37.30	64.33	87.12
Qwen2.5-VL-7B-Instruct	7.47	6.72	0.70	20.41	49.49	70.00
Llama-3.1-70B-Instruct	18.67	18.04	4.90	33.36	59.56	88.64
Llama-3.1-8B-Instruct	11.33	10.16	3.50	22.75	65.42	80.00



Reference Snapshot Generated Snapshot

CLIP Score: 75.54 VLM-judge Score: 97.14

(a) A Huffman Tree Encoding demonstration.



Reference Snapshot Generated Snapshot

CLIP Score: 82.74 VLM-judge Score: 32.00

(b) A Spring-Mass-Damper System demonstration.

Figure 4. Examples comparing CLIP and VLM-Judge scores.

- ***The Leaderboard***: Proprietary frontier models (e.g., GPT-5, Claude-Sonnet) lead, but even the best scores hover around 55-60%.
- ***Key Finding***: Models exhibit a massive gap between UI Success Rate (>80%) and Scientific Correctness (<50%).
- ***Failure Taxonomy***:
  - 53% Logic Errors: Disconnect between UI state and scientific equations.
  - 42% Rendering Issues: Failures in canvas library integrations (e.g., p5.js).

Potential Impact & Takeaways

- ***Standardizing Evaluation***

- Shifts the community from evaluating **“code that looks right”** to **“code that interacts correctly”**. — a critical safeguard for pedagogical rigor.

- ***Future AI for Science and Education***

- Evaluation attempt for building autonomous educational tools and interactive scientific assistants.

- Empowering teachers & students to deeply understand, internalize, and apply knowledge.

- ***Call to Action***

- Code, benchmark, results are all fully open-sourced at GitHub

https://github.com/open-compass/InteractScience

E-mail: qschen@smail.nju.edu.cn

Homepage ***Github***

Feel free to reach out!