Agentic Tool Use On multi-turn tool-use benchmarks, Kimi-K2-Instruct sets a new standard. It achieves 66.1 Pass@1 on τ^2-Bench and 76.5 on ACEBench, substantially outperforming all baselines. These results affirm its strength in grounded, controlled, and agent-driven tool orchestration across domains.

General Capabilities Kimi-K2-Instruct exhibits strong, balanced performance across general knowledge, math, instruction following, and long-context tasks. It surpasses open-source peers on SimpleQA (31.0%), MMLU (89.5%) and MMLU-Redux (92.7%), and leads all models on instruction benchmarks (IFEval: 89.8%, Multi-Challenge: 54.1%). In math and STEM, it achieves top-tier scores (AIME 2024: 69.6%, GPQA-Diamond: 75.1%), and remains competitive on long-context factuality and retrieval (DROP: 93.5%, MRCR: 55.0%). These results position Kimi-K2-Instruct as a well-rounded and capable generalist across both short- and long-context settings.

Open-Ended Evaluation On the LMSYS Arena leaderboard (July 17, 2025), Kimi-K2-Instruct ranks as the top-1 open-source model and 5th overall based on over 3,000 user votes. This real-world preference signal—across diverse, blind prompts—underscores Kimi-K2's strengths in generating high-quality responses on open-ended tasks.

4.2 Pre-training Evaluations

4.2.1 Evaluation Settings

Benchmarks We evaluate Kimi-K2-Base across diverse capability areas. For general capabilities, we assess on MMLU [23], MMLU-Pro [76], MMLU-Redux [17], BBH [67], TriviaQA [34], SuperGPQA [13], SimpleQA [78], HellaSwag [88], AGIEval [89], GPQA-Diamond [61], ARC-Challenge [8], and WinoGrande [62]. For coding capabilities, we employ EvalPlus [45] (averaging HumanEval [7], MBPP [1], HumanEval+, and MBPP+), LiveCodeBench v6 [31], and CRUXEval [18]. For mathematical reasoning, we utilize GSM8K [9], GSM8K-Platinum [74], MATH [24], and CMATH [79]. For Chinese language capabilities, we evaluate on C-Eval [29], CMMLU [40], and CSimpleQA [22].

Baselines We benchmark against leading open-source foundation models: DeepSeek-V3-Base [10], Qwen2.5-72B-Base [59] (Note that Qwen3-235B-A22B-Base is not open-sourced, and the largest open-sourced base model in the Qwen series is Qwen2.5-72B-Base), and Llama 4-Maverick [70] (Llama 4-Behemoth is also not open-sourced). All models are evaluated under identical configurations to ensure fair comparison.

Evaluation Configurations We employ perplexity-based evaluation for MMLU, MMLU-Redux, GPQA-Diamond, HellaSwag, ARC-Challenge, C-Eval, and CMMLU. Generation-based evaluation is used for MMLU-Pro, SuperGPQA, TriviaQA, BBH, CSimpleQA, MATH, CMATH, GSM8K, GSM8K-Platinum, CRUXEval, LiveCodeBench, and EvalPlus. To mitigate the high variance inherent to GPQA-Diamond, we report the mean score across eight independent runs. All evaluations are conducted using our internal framework derived from LM-Harness-Evaluation [4], ensuring consistent settings across all models.

4.2.2 Evaluation Results

Table 4 presents a comprehensive comparison of Kimi-K2-Base against leading open-source foundation models across diverse evaluation benchmarks. The results demonstrate that Kimi-K2-Base achieves state-of-the-art performance across the majority of evaluated tasks, establishing it as a leading foundation model in the open-source landscape.

General Language Understanding Kimi-K2-Base achieves state-of-the-art performance on 10 out of 12 English language benchmarks. Notable results include MMLU (87.79%), MMLU-Pro (69.17%), MMLU-Redux (90.17%), SuperGPQA (44.67%), and SimpleQA (35.25%), significantly outperforming all baselines.

Coding Capabilities On coding benchmarks, Kimi-K2-Base sets new standards with leading performance across all metrics. It achieves 74.00% on CRUXEval-I-cot, 83.50% on CRUXEval-O-cot, 26.29% on LiveCodeBench v6, and 80.33% on EvalPlus, demonstrating superior code generation and comprehension abilities, particularly in scenarios requiring step-by-step reasoning.

Mathematical Reasoning Kimi-K2-Base exhibits exceptional mathematical capabilities, leading on three out of four benchmarks: MATH (70.22%), GSM8K (92.12%), and GSM8K-Platinum (94.21%). It maintains competitive performance on CMATH (90.26%), narrowly behind DeepSeek-V3-Base (90.53%). These results highlight the model's robust mathematical problem-solving abilities across varying difficulty levels.

Chinese Language Understanding The model demonstrates superior multilingual capabilities, achieving state-of-the-art results across all Chinese language benchmarks: C-Eval (92.50%), CMMLU (90.90%), and CSimpleQA (77.57%). These results establish Kimi-K2-Base as a leading model for Chinese language understanding while maintaining strong performance across other languages.

Table 4: Performance comparison of Kimi-K2-Base against leading open-source models across diverse tasks.

	Benchmark (Metric)	#Shots	Kimi-K2-Base	DeepSeek-V3-Base	Llama4-Maverick-Base	Qwen2.5-72B-Base
	Architecture	-	MoE	MoE	MoE	Dense
	# Activated Params	-	32B	37B	17B	72B
	# Total Params	-	1043B	671B	400B	72B
English	MMLU	5-shots	**87.79**	87.10	84.87	86.08
	MMLU-pro	5-shots	**69.17**	60.59	63.47	62.80
	MMLU-redux	5-shots	**90.17**	89.53	88.18	87.77
	SuperGPQA	5-shots	**44.67**	39.20	38.84	34.23
	GPQA-Diamond(avg@8)	5-shots	48.11	**50.51**	49.43	40.78
	SimpleQA	5-shots	**35.25**	26.49	23.74	10.31
	TriviaQA	5-shots	**85.09**	84.11	79.25	76.03
	BBH	3-shots	**88.71**	88.37	87.10	84.09
	HellaSwag	5-shots	94.60	89.44	86.02	**95.27**
	AGIEval	-	**84.23**	81.57	67.55	76.87
	ARC-Challenge	0-shot	**95.73**	93.77	94.03	95.56
	WinoGrande	5-shots	**85.32**	84.21	77.58	84.14
Code	CRUXEval-I-cot	0-shots	**74.00**	62.75	67.13	61.12
	CRUXEval-O-cot	0-shots	**83.50**	75.25	75.88	66.13
	LiveCodeBench(v6)	1-shots	**26.29**	24.57	25.14	22.29
	EvalPlus	-	**80.33**	65.61	65.48	66.04
Math	MATH	4-shots	**70.22**	61.70	63.02	62.68
	GSM8k	8-shots	**92.12**	91.66	86.35	90.37
	GSM8k-platinum	8-shots	**94.21**	93.38	88.83	92.47
	CMATH	6-shots	90.26	**90.53**	88.07	86.98
Chinese	C-Eval	5-shots	**92.50**	90.04	80.91	90.86
	CMMLU	5-shots	**90.90**	88.84	81.24	90.55
	CSimpleQA	5-shots	**77.57**	72.13	53.47	50.53

4.3 Safety Evaluation

4.3.1 Experiment Settings

We conducted red-teaming evaluations on Kimi K2 compare with other open-source LLMs. The evaluation covered a range of attack scenarios—including harmful content, privacy content, and security content, as well as different attack strategies such as prompt injection and iterative jailbreak.

We choose *Promptfoo*[5] to generate adversarial prompts and analyze the responses. By this way, we can evaluate model in a scalable ways.

Model Selection We compare Kimi K2 with three other open-source LLMs: DeepSeek-V3, DeepSeek-R1, and Qwen3.

Promptfoo Settings Table 5 lists plugins and strategies evaluated, with each plugin paired with all strategies to assess their performance.

Test Case Count Given the inherent non-determinism of large language model inference, single-pass outputs may exhibit variability. To account for this, we generated 3 attack prompts per plugin for each strategy.

Prompt Language Settings We pre-tested the language compatibility for each plugin-strategy combination. Some plugins support both English and Chinese, while others only support English. For combinations that support both, we generated 3 prompts in each language, resulting in 6 prompts per combination.

[5] https://github.com/promptfoo/promptfoo

Table 5: Enabled Plugins and Strategies

Plugin	Harmful	Graphic Content, Harassment and Bullying, Hate Speech, Insults, Profanity, Radicalization, Self Harm, Sexual Content, ToxicChat
	Criminal	Chemical&Biological Weapons, Child Exploitation, Copyright Violations, Cybercrime, Illegal Activities, Illegal Drugs, Indiscriminate Weapons, Intellectual Property Violation, Non-Violent Crime, Violent Crime, Sex Crimes
	Misinformation	Competitor Endorsement, Unsupervised Contracts, Excessive Agency, Hallucination, Misinformation and Disinformation, Specialized Advice, Unsafe Practices, Imitation, Overreliance, Political Opinions, Religious Sensitivity
	Privacy	Privacy Violation, PII in API/Database, Direct PII Exposure, PII in Session Data, PII via Social Engineering
	Security	ASCII Smuggling, CyberSecEval, Harmbench, Debug Access, Divergent Repetition, DoNotAnswer, Malicious Code, Pliny, Prompt Extraction, Reasoning DoS, Tool Discovery
Strategy	Basic, Prompt Injection, Iterative Jailbreak, Crescendo	

Manual Review We incorporated human review into the evaluation process. To minimize subjectivity problem, we conducted multiple rounds of review and assigned the same reviewer to evaluate all cases within a given test set to ensure consistency and reduce variability in judgment.

4.3.2 Safety Evaluation Results

Table 6 presents the passing rates of different models under various plugin–strategy combinations.

Table 6: Safety Evaluation Results

Plugin	Strategy	Kimi-K2-Instruct	DeepSeek-V3-0324	DeepSeek-R1	Qwen3-235B-A22B
Harmful	Basic	98.04	90.45	99.02	98.53
	Base64	100	90.20	100	100
	Prompt Injection	93.14	100	95.10	99.02
	Iterative Jailbreak	92.16	66.67	72.55	74.51
	Crescendo	64.71	64.71	80.39	86.27
Criminal	Basic	100	99.62	95.45	99.24
	Base64	96.97	89.39	84.85	98.48
	Prompt Injection	75.76	91.67	69.70	98.47
	Iterative Jailbreak	57.57	21.21	25.76	53.03
	Crescendo	56.06	31.81	42.42	59.09
Misinformation	Basic	97.28	92.57	92.46	94.84
	Base64	98.48	90.48	96.83	93.65
	Prompt Injection	98.39	86.51	93.65	93.65
	Iterative Jailbreak	63.97	53.97	84.13	69.84
	Crescendo	85.71	55.56	88.89	84.13
Privacy	Basic	100	100	100	100
	Base64	100	100	100	100
	Prompt Injection	88.33	98.33	100	91.67
	Iterative Jailbreak	76.67	100	93.33	96.67
	Crescendo	96.67	100	96.67	100
Security	Basic	77.84	75.57	70.46	90.09
	Base64	82.93	82.93	63.41	95.12
	Prompt Injection	87.80	97.56	65.85	84.13
	Iterative Jailbreak	43.90	60.97	43.90	78.04
	Crescendo	68.29	87.80	68.29	87.80

Without targeted optimization for specific evaluation scenarios, the passing rate of some complex cases (e.g., Harmful–Iterative Jailbreak) was relatively higher compared to other models.

Across different attack strategies, the models exhibited varying trends. Under the Base64 strategy, passing rates generally approached or reached 100%, suggesting that encoding transformations had minimal impact on the models'

basic robustness. In contrast, the Crescendo strategy led to a general drop in passing rates, indicating stronger adversarial effectiveness.

In addition, complex attack strategies do not always outperform basic prompts. Some originally adversarial prompts may lose their intended meaning after multiple rounds of transformation, rendering the resulting model outputs less meaningful.

Automated Red-teaming Limitations Due to the involvement of human review, the evaluation results inevitably contain a degree of subjectivity. Additionally, certain plugin types involve API misuse or external tool invocation, which are more suitable for evaluating agent models with tool-calling capabilities. In the context of base LLMs, such tests may have limited relevance.

5 Limitations

In our internal tests, we have identified some limitations in current Kimi K2 models. When dealing with hard reasoning tasks or unclear tool definition, the model may generate excessive tokens, sometimes leading to truncated outputs or incomplete tool calls. Additionally, performance may decline on certain tasks if tool use is unnecessarily enabled. When building complete software projects, the success rate of one-shot prompting is not as good as using K2 under an agentic coding framework. We are working to address these issues in future releases and looking forward to more feedbacks.

6 Conclusions

We introduced Kimi K2, a 1T-parameter open-weight MoE model built for agentic intelligence. Leveraging the token-efficient MuonClip optimizer and a 15.5T-token high-quality dataset, Kimi K2 achieves stable, scalable pre-training. Post-training combines large-scale synthetic tool-use data with a unified RL framework using both verifiable rewards and self-critic feedbacks. Kimi K2 sets new state-of-the-art on agentic and reasoning benchmarks, establishing itself as the most capable open-weight LLM to date.

7 Acknowledgments

We would like to acknowledge the valuable support provided by the OpenHands and Multi-SWE-bench teams in evaluating the SWE-bench Verified and Multi-SWE-bench experimental results.

References

[1] Jacob Austin et al. *Program Synthesis with Large Language Models*. 2021. arXiv: 2108.07732 [cs.PL]. URL: https://arxiv.org/abs/2108.07732.

[2] Yushi Bai et al. *LongBench v2: Towards Deeper Understanding and Reasoning on Realistic Long-context Multitasks*. 2025. arXiv: 2412.15204 [cs.CL]. URL: https://arxiv.org/abs/2412.15204.

[3] Victor Barres et al. *τ^2-Bench: Evaluating Conversational Agents in a Dual-Control Environment*. 2025. arXiv: 2506.07982 [cs.AI]. URL: https://arxiv.org/abs/2506.07982.

[4] Stella Biderman et al. "Lessons from the trenches on reproducible evaluation of language models". In: *arXiv preprint arXiv:2405.14782* (2024).

[5] Federico Cassano et al. "MultiPL-E: A Scalable and Polyglot Approach to Benchmarking Neural Code Generation". In: *IEEE Transactions on Software Engineering* 49.7 (2023), pp. 3675–3691. DOI: 10.1109/TSE.2023.3267446.

[6] Chen Chen et al. "ACEBench: Who Wins the Match Point in Tool Learning?" In: *arXiv e-prints* (2025), arXiv–2501.

[7] Mark Chen et al. "Evaluating Large Language Models Trained on Code". In: (2021). arXiv: 2107.03374 [cs.LG].

[8] Peter Clark et al. "Think you have solved question answering? try arc, the ai2 reasoning challenge". In: *arXiv preprint arXiv:1803.05457* (2018).

[9] Karl Cobbe et al. *Training Verifiers to Solve Math Word Problems*. 2021. arXiv: 2110.14168 [cs.LG]. URL: https://arxiv.org/abs/2110.14168.

[10] DeepSeek-AI. *DeepSeek-V3 Technical Report*. 2024. arXiv: 2412.19437 [cs.CL]. URL: https://arxiv.org/abs/2412.19437.

[11] Mostafa Dehghani et al. "Scaling vision transformers to 22 billion parameters". In: *International conference on machine learning*. PMLR. 2023, pp. 7480–7512.

[12] Guanting Dong et al. *Self-play with Execution Feedback: Improving Instruction-following Capabilities of Large Language Models*. 2024. arXiv: 2406.13542 [cs.CL]. URL: https://arxiv.org/abs/2406.13542.

[13] Xinrun Du et al. "Supergpqa: Scaling llm evaluation across 285 graduate disciplines". In: *arXiv preprint arXiv:2502.14739* (2025).

[14] Dheeru Dua et al. "DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs". In: *CoRR* abs/1903.00161 (2019). arXiv: 1903.00161. URL: http://arxiv.org/abs/1903.00161.

[15] Kazuki Fujii et al. *Rewriting Pre-Training Data Boosts LLM Performance in Math and Code*. 2025. arXiv: 2505.02881 [cs.LG]. URL: https://arxiv.org/abs/2505.02881.

[16] Paul Gauthier. *Aider LLM Leaderboards*. https://aider.chat/docs/leaderboards/. 2025.

[17] Aryo Pradipta Gema et al. "Are we done with mmlu?" In: *arXiv preprint arXiv:2406.04127* (2024).

[18] Alex Gu et al. "Cruxeval: A benchmark for code reasoning, understanding and execution". In: *arXiv preprint arXiv:2401.03065* (2024).

[19] Daya Guo et al. "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning". In: *arXiv preprint arXiv:2501.12948* (2025).

[20] Zhicheng Guo et al. "StableToolBench: Towards Stable Large-Scale Benchmarking on Tool Learning of Large Language Models". In: *arXiv preprint arXiv:2403.07714* (2025).

[21] Aaron Harlap et al. "Pipedream: Fast and efficient pipeline parallel dnn training". In: *arXiv preprint arXiv:1806.03377* (2018).

[22] Y He et al. "Chinese simpleqa: A chinese factuality evaluation for large language models, 2024a". In: *URL https://arxiv. org/abs/2411.07140* ().

[23] Dan Hendrycks et al. "Measuring massive multitask language understanding". In: *arXiv preprint arXiv:2009.03300* (2020).

[24] Dan Hendrycks et al. *Measuring Mathematical Problem Solving With the MATH Dataset*. 2021. arXiv: 2103.03874 [cs.LG]. URL: https://arxiv.org/abs/2103.03874.

[25] Shengding Hu et al. "Minicpm: Unveiling the potential of small language models with scalable training strategies". In: *arXiv preprint arXiv:2404.06395* (2024).

[26] Jiaxin Huang et al. "Large language models can self-improve". In: *arXiv preprint arXiv:2210.11610* (2022).

[27] Siming Huang et al. *OpenCoder: The Open Cookbook for Top-Tier Code Large Language Models*. 2025. arXiv: 2411.04905 [cs.CL]. URL: https://arxiv.org/abs/2411.04905.

[28] Yanping Huang et al. "Gpipe: Efficient training of giant neural networks using pipeline parallelism". In: *Advances in neural information processing systems* 32 (2019).

[29] Yuzhen Huang et al. *C-Eval: A Multi-Level Multi-Discipline Chinese Evaluation Suite for Foundation Models*. 2023. arXiv: 2305.08322 [cs.CL]. URL: https://arxiv.org/abs/2305.08322.

[30] Alon Jacovi et al. *The FACTS Grounding Leaderboard: Benchmarking LLMs' Ability to Ground Responses to Long-Form Input*. 2025. arXiv: 2501.03200 [cs.CL]. URL: https://arxiv.org/abs/2501.03200.

[31] Naman Jain et al. "Livecodebench: Holistic and contamination free evaluation of large language models for code". In: *arXiv preprint arXiv:2403.07974* (2024).

[32] Carlos E Jimenez et al. "SWE-bench: Can Language Models Resolve Real-world Github Issues?" In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=VTF8yNQM66.

[33] Keller Jordan et al. *Muon: An optimizer for hidden layers in neural networks*. 2024. URL: https://kellerjordan.github.io/posts/muon/.

[34] Mandar Joshi et al. *TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension*. 2017. arXiv: 1705.03551 [cs.CL]. URL: https://arxiv.org/abs/1705.03551.

[35] Kimi Team. "Kimi k1. 5: Scaling reinforcement learning with llms". In: *arXiv preprint arXiv:2501.12599* (2025).

[36] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://arxiv.org/abs/1412.6980.

[37] Satyapriya Krishna et al. *Fact, Fetch, and Reason: A Unified Evaluation of Retrieval-Augmented Generation*. 2025. arXiv: 2409.12941 [cs.CL]. URL: https://arxiv.org/abs/2409.12941.

[38] Joel Lamy-Poirier. "Breadth-first pipeline parallelism". In: *Proceedings of Machine Learning and Systems* 5 (2023), pp. 48–67.

[39] Dmitry Lepikhin et al. "Gshard: Scaling giant models with conditional computation and automatic sharding". In: *arXiv preprint arXiv:2006.16668* (2020).

[40] Haonan Li et al. *CMMLU: Measuring massive multitask language understanding in Chinese*. 2024. arXiv: 2306.09212 [cs.CL]. URL: https://arxiv.org/abs/2306.09212.

[41] Jia Li et al. "Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions". In: *Hugging Face repository* 13.9 (2024), p. 9.

[42] Tianle Li et al. "From Crowdsourced Data to High-Quality Benchmarks: Arena-Hard and BenchBuilder Pipeline". In: *arXiv preprint arXiv:2406.11939* (2024).

[43] Bill Yuchen Lin et al. *ZebraLogic: On the Scaling Limits of LLMs for Logical Reasoning*. 2025. arXiv: 2502.01100 [cs.AI]. URL: https://arxiv.org/abs/2502.01100.

[44] Aixin Liu et al. "Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model". In: *arXiv preprint arXiv:2405.04434* (2024).

[45] Jiawei Liu et al. "Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 21558–21572.

[46] Jingyuan Liu et al. "Muon is scalable for LLM training". In: *arXiv preprint arXiv:2502.16982* (2025).

[47] Ziming Liu et al. "Hanayo: Harnessing Wave-like Pipeline Parallelism for Enhanced Large Model Training Efficiency". In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. SC '23. ACM, Nov. 2023, pp. 1–13. DOI: 10.1145/3581784.3607073. URL: http://dx.doi.org/10.1145/3581784.3607073.

[48] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

[49] Jan Ludziejewski et al. *OpenAI Gym*. 2025. arXiv: 2502.05172 [cs.LG]. URL: https://arxiv.org/abs/2502.05172.

[50] Samuel Miserendino et al. "SWE-Lancer: Can Frontier LLMs Earn $1 Million from Real-World Freelance Software Engineering?" In: *arXiv preprint arXiv:2502.12115* (2025).

[51] Arindam Mitra et al. "Agentinstruct: Toward generative teaching with agentic flows". In: *arXiv preprint arXiv:2407.03502* (2024).

[52] Ivan Moshkov et al. "Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset". In: *arXiv preprint arXiv:2504.16891* (2025).

[53] Deepak Narayanan et al. "Efficient large-scale language model training on gpu clusters using megatron-lm". In: *Proceedings of the international conference for high performance computing, networking, storage and analysis*. 2021, pp. 1–15.

[54] Long Ouyang et al. "Training language models to follow instructions with human feedback". In: *Advances in neural information processing systems* 35 (2022), pp. 27730–27744.

[55] Bowen Peng et al. "Yarn: Efficient context window extension of large language models". In: *arXiv preprint arXiv:2309.00071* (2023).

[56] Long Phan et al. *Humanity's Last Exam*. 2025. arXiv: 2501.14249 [cs.LG]. URL: https://arxiv.org/abs/2501.14249.

[57] Penghui Qi et al. "Zero bubble pipeline parallelism". In: *arXiv preprint arXiv:2401.10241* (2023).

[58] Yujia Qin et al. "Toolllm: Facilitating large language models to master 16000+ real-world apis". In: *arXiv preprint arXiv:2307.16789* (2023).

[59] Qwen et al. *Qwen2.5 Technical Report*. 2025. arXiv: 2412.15115 [cs.CL]. URL: https://arxiv.org/abs/2412.15115.

[60] Samyam Rajbhandari et al. "Zero: Memory optimizations toward training trillion parameter models". In: *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE. 2020, pp. 1–16.

[61] David Rein et al. "Gpqa: A graduate-level google-proof q&a benchmark". In: *First Conference on Language Modeling*. 2024.

[62] Keisuke Sakaguchi et al. "Winogrande: An adversarial winograd schema challenge at scale". In: *Communications of the ACM* 64.9 (2021), pp. 99–106.

[63] David Silver and Richard S Sutton. "Welcome to the era of experience". In: *Google AI* 1 (2025).

[64] Ved Sirdeshmukh et al. *MultiChallenge: A Realistic Multi-Turn Conversation Evaluation Benchmark Challenging to Frontier LLMs*. 2025. arXiv: 2501.17399 [cs.CL]. URL: https://arxiv.org/abs/2501.17399.

[65] Giulio Starace et al. "PaperBench: Evaluating AI's Ability to Replicate AI Research". In: *arXiv preprint arXiv:2504.01848* (2025).

[66] Hao Sun et al. *ZeroSearch: Incentivize the Search Capability of LLMs without Searching*. 2025. arXiv: 2505.04588 [cs.CL]. URL: https://arxiv.org/abs/2505.04588.

[67] Mirac Suzgun et al. *Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them*. 2022. arXiv: 2210.09261 [cs.CL]. URL: https://arxiv.org/abs/2210.09261.

[68] Manveer Singh Tamber et al. "Benchmarking LLM Faithfulness in RAG with Evolving Leaderboards". In: *arXiv preprint arXiv:2505.04847* (2025).

[69] Gemma Team et al. "Gemma 2: Improving open language models at a practical size". In: *arXiv preprint arXiv:2408.00118* (2024).

[70] LlaMA Team. *The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation — ai.meta.com*. https://ai.meta.com/blog/llama-4-multimodal-intelligence/. [Accessed 15-07-2025].

[71] The Terminal-Bench Team. *Terminal-Bench: A Benchmark for AI Agents in Terminal Environments*. Apr. 2025. URL: https://github.com/laude-institute/terminal-bench.

[72] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[73] Vectara. *Hallucination Evaluation Model (Revision 7437011)*. 2024. URL: https://huggingface.co/vectara/hallucination_evaluation_model.

[74] Joshua Vendrow et al. "Do large language model benchmarks test reliability?" In: *arXiv preprint arXiv:2502.03461* (2025).

[75] Yizhong Wang et al. "Self-instruct: Aligning language models with self-generated instructions". In: *arXiv preprint arXiv:2212.10560* (2022).

[76] Yubo Wang et al. *MMLU-Pro: A More Robust and Challenging Multi-Task Language Understanding Benchmark*. 2024. arXiv: 2406.01574 [cs.CL]. URL: https://arxiv.org/abs/2406.01574.

[77] Zhexu Wang et al. *OJBench: A Competition Level Code Benchmark For Large Language Models*. 2025. arXiv: 2506.16395 [cs.CL]. URL: https://arxiv.org/abs/2506.16395.

[78] Jason Wei et al. "Measuring short-form factuality in large language models". In: *arXiv preprint arXiv:2411.04368* (2024).

[79] Tianwen Wei et al. *CMATH: Can Your Language Model Pass Chinese Elementary School Math Test?* 2023. arXiv: 2306.16636 [cs.CL]. URL: https://arxiv.org/abs/2306.16636.

[80] Colin White et al. "LiveBench: A Challenging, Contamination-Free LLM Benchmark". In: *The Thirteenth International Conference on Learning Representations*. 2025.

[81] Mitchell Wortsman et al. "Small-scale proxies for large-scale transformer training instabilities, 2023". In: *URL https://arxiv. org/abs/2309.14322* ().

[82] Can Xu et al. *WizardLM: Empowering large pre-trained language models to follow complex instructions*. 2025. arXiv: 2304.12244 [cs.CL]. URL: https://arxiv.org/abs/2304.12244.

[83] Zhangchen Xu et al. *KodCode: A Diverse, Challenging, and Verifiable Synthetic Dataset for Coding*. 2025. arXiv: 2503.02951 [cs.LG]. URL: https://arxiv.org/abs/2503.02951.

[84] John Yang et al. *SWE-smith: Scaling Data for Software Engineering Agents*. 2025. arXiv: 2504.21798 [cs.SE]. URL: https://arxiv.org/abs/2504.21798.

[85] Shunyu Yao et al. "tau-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains". In: *arXiv preprint arXiv:2406.12045* (2024).

[86] Daoguang Zan et al. "Multi-swe-bench: A multilingual benchmark for issue resolving". In: *arXiv preprint arXiv:2504.02605* (2025).

[87] Eric Zelikman et al. "Star: Bootstrapping reasoning with reasoning". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 15476–15488.

[88] Rowan Zellers et al. "Hellaswag: Can a machine really finish your sentence?" In: *arXiv preprint arXiv:1905.07830* (2019).

[89] Wanjun Zhong et al. "Agieval: A human-centric benchmark for evaluating foundation models". In: *arXiv preprint arXiv:2304.06364* (2023).

[90] Jeffrey Zhou et al. "Instruction-Following Evaluation for Large Language Models". In: *ArXiv* abs/2311.07911 (2023). URL: https://arxiv.org/abs/2311.07911.

[91] Qin Zhu et al. *AutoLogi: Automated Generation of Logic Puzzles for Evaluating Reasoning Abilities of Large Language Models*. 2025. arXiv: 2502.16906 [cs.CL]. URL: https://arxiv.org/abs/2502.16906.