Delphi Model Evaluation Report: General Reasoning in Middleweight LLMs

Gensyn AI Team

We track the performance of “middleweight” variants of open source LLMs published by popular frontier labs. Over three weeks we will run evaluations across 11 general AI reasoning benchmarks that have been curated to capture broad real-world reasoning ability.

Entrant models

We evaluate the following models, as provided on HuggingFace, using the default datatype provided and versions current as of December 7, 2025.

- Qwen3-30B-A3B-Instruct-2507 from Qwen.
- gpt-oss-20b from OpenAI.
- gemma-3-27b-it from Google.
- GLM-4-32B-0414 from Zai-Org.
- Falcon-H1-34B-Instruct from TII-UAE.

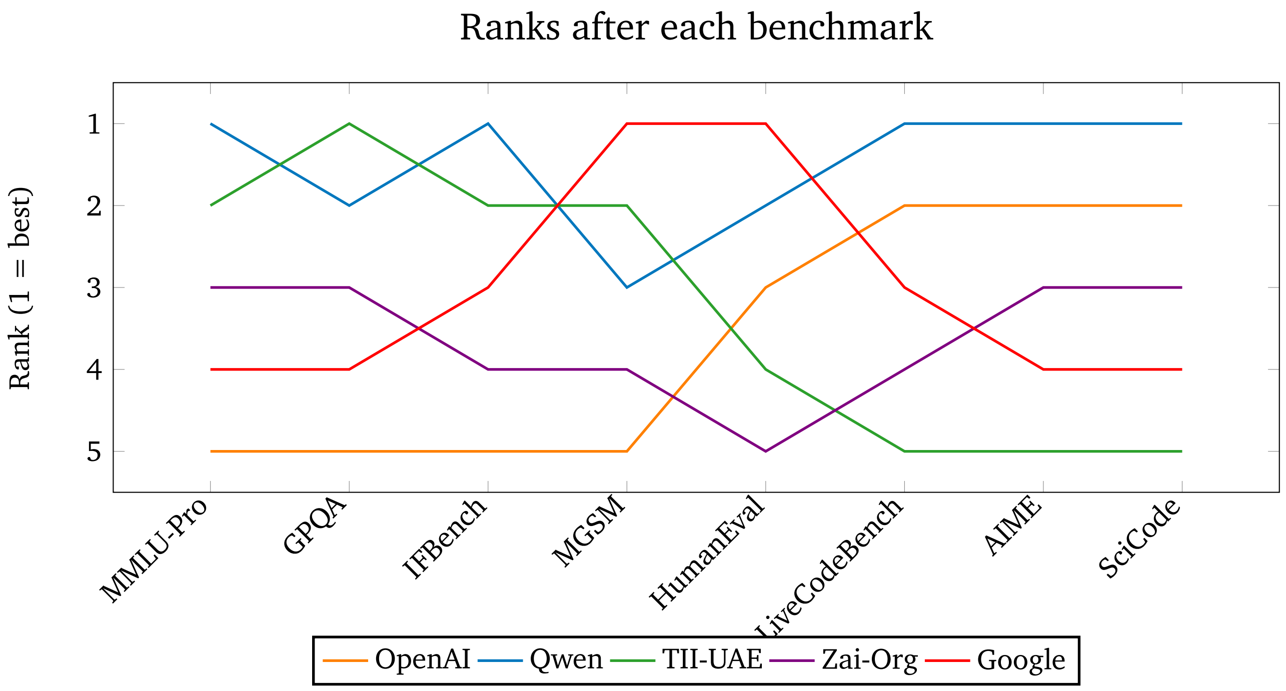
Evaluation setup: We perform inference with temperature 0.0 on machines with NVIDIA H100 GPUs using vLLM, unless specified.

Overall scores

The running average after 8 evaluations.

Rank	Family	Model	Avg. score	Δ rank	Notes
1	Qwen	Qwen3-30B-A3B-Instruct-2507	56.42	-	-
2	OpenAI	gpt-oss-20b	55.62	-	-
3	Zai-Org	GLM-4-32B-0414	50.56	-	-
4	Google	gemma-3-27b-it	49.37	-	-
5	TIIUAE	Falcon-H1-34B-Instruct	48.45	-	-

Avg. score is the mean of normalized benchmark scores. Δ *rank* is the change in rank compared to the last evaluation.



Per-Benchmark Breakdown

Thus far we have revealed 8 out of 11 evaluations

Benchmark	gpt-oss-20b	Qwen3-30B-A3B-Instruct-2507	Falcon-H1-34B-Instruct	GLM-4-32B-0414	gemma-3-27b-it
MMLU-Pro (17)	47.88	74.76	72.27	67.52	65.19
GPQA-Diamond (14)	37.22	46.92	49.60	46.97	45.66
IFBench (13)	54.56	34.83	32.59	31.16	40.48
MGSM (15)	77.16	70.40	73.56	73.49	88.25
HumanEval (15)	90.90	85.40	76.80	79.90	75.60
LiveCodeBench (12)	51.33	47.77	36.24	46.30	32.12
AIME	55.33	60.67	18.33	32.33	26.00
SciCode (16)	30.58	30.58	28.18	26.80	21.65
?	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
?	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
?	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

Rows with "?" as the benchmark will be revealed in the near future as we run those evaluations. Stay tuned for more.

Benchmark 8: SciCode

The SciCode benchmark (1) evaluates code generation capabilities for solving realistic scientific research problems. It consists of 80 main problems decomposed into 338 subproblems, covering 16 subdomains from 6 domains: Physics, Math, Material Science, Biology, and Chemistry. Problems are derived from real scientists' workflows and focus on numerical methods, system simulations, and scientific calculations. Unlike exam-style benchmarks, SciCode requires models to demonstrate deep scientific knowledge, reasoning, and code synthesis abilities. A main problem is considered solved only if all its subproblems pass the scientist-annotated test cases.

Experimental setup. We evaluate all 65 problems from the SciCode test split (288 subproblems). Models are evaluated in zero-shot mode with scientist-annotated background knowledge provided (`with_background=True`), which supplies the necessary scientific context and reasoning steps. This setting shifts the evaluation focus towards coding and instruction-following capabilities. Results are presented as subproblem accuracy. Note that temperature is set to 0 for deterministic outputs.

We make the following observations:

- Qwen3-30B-A3B-Instruct-2507 and gpt-oss-20b tied for highest subproblem accuracy at 30.58%. Note that Qwen3 solved fewer main problems (6.2%) and gpt-oss-20b solved two problems that no other model solved.

- Google's gemma-3-27b-it continues struggling with coding, solving 0 main problems and achieving a subproblem accuracy of 21.9%.

Benchmark 7: American Invitational Mathematics Examination (AIME)

The "American Invitational Mathematics Examination" is a competition-level dataset of math problems targeted at high school students. There are two parts, each composed of 15 questions; we evaluate our models on all 30 questions. Questions in this dataset are scored using a judge model to parse answers and semantically compare them to a given ground truth answer. The dataset can be accessed here: HuggingFace:math-ai/aime25. Frontier models perform well on the dataset, with GPT-5, for example, achieving 99% (6) without tools, such as a Python interpreter.

Experimental setup. We run each model 10 times on each question, sampling with temperature 0.7, top-p 0.95, and report the average accuracy across the 10 runs. We use gpt-5-mini as the scoring model, adapting the code from Humanity's Last Exam (3). The models were given a budget of 12,288 output tokens (but the models usually finished in a few thousand tokens). We also tested various system prompts and chose the best accuracy for each model among the prompts.

We make the following observations:

- This dataset showed extreme variance between models, with Qwen3-30B-A3B-Instruct-2507 and gpt-oss-20b scoring above 55% while the rest scored between 18% and 32%.

- While gemma-3-27b-it significantly outperformed the other models on the previous mathematics benchmark (Benchmark 4: MGSM), it scored second-to-last at 26%.

Benchmark 6: LiveCodeBench

LiveCodeBench (12) is a large-scale, execution-based benchmark designed to evaluate code generation models on realistic competitive programming tasks. It consists of several hundred problems sourced from platforms such as Codeforces, LeetCode, and AtCoder, spanning a wide range of algorithmic domains, including data structures, graph algorithms, dynamic programming, and number theory. Solutions are assessed by compiling and executing generated code against test cases under strict correctness constraints. Note that while LiveCodeBench has a time-ordered dataset to help prevent contamination with a model's own training data, we evaluate models on the full set of problems. After preventing contamination, frontiers models like Gemini 3 score just above 90% (2).

Experimental setup. We evaluate all 1055 problems from LiveCodeBench (5). We perform pass@1 evaluation with the repository's default settings: temperature=0.2, and maximum output length of 2048 tokens.

We make the following observations:

- OpenAI's gpt-oss-20b continues its strong performance in coding benchmarks, achieving the highest score at about 51%.

- On the flip side, Google's gemma-3-27b-it struggles the most with coding, scoring the lowest at 32%. Interestingly, gemma-3-27b-it also improves only negligibly when moving from pass@1 to pass@5 and pass@10 scoring.

Benchmark 5: HumanEval

The HumanEval benchmark (10) assesses code generation capabilities by making models complete Python function implementations given a function signature and docstring. It consists of 164 handwritten programming problems covering fundamental algorithms, data structures, string manipulation, and mathematical operations. We use the EvalPlus framework(11), which extends the original benchmark with comprehensive test suites containing approximately 80 times more test cases per problem to rigorously evaluate both functional correctness against edge cases. HumanEval is a valuable benchmark as its compact, specification-based challenges effectively measure how well a model can convert natural language descriptions into working code. It offers clear insights into its reasoning capabilities and ability to generalize when generating programs.

Experimental setup. We evaluate all 164 problems from HumanEval, using the EvalPlus library (4), which extends the original benchmark with comprehensive test suites. For pass@k evaluation, we generate k=10 samples per problem with temperature=0.2, yielding 1,640 total samples. Results are presented as pass@10 metrics. Note that the models are given a maximum output length of 2048 tokens.

We make the following observations:

- OpenAI's gpt-oss-20b scored the highest, with a pass@10 score of 90%. Note that even frontier models, such as GPT-5 and Claude 3.5 Sonnet, score under 95%.

- The remaining models all scored between 75% and 85%.

Benchmark 4: Multilingual Grade School Math Benchmark (MGSM)

The Multilingual Grade School Math (MGSM) benchmark (15) assesses mathematical reasoning abilities across eleven languages. It is built from 250 elementary-level math word problems drawn from the English GSM8K dataset (9), which were professionally translated into Bengali, Chinese, French, German, Japanese, Russian, Spanish, Swahili, Telugu, and Thai. The tasks involve multi-step arithmetic reasoning that a middle-schooler should be able to solve.

Experimental setup. We evaluate all 250 problems in each of the eleven languages, yielding 2,750 total instances, and score performance using exact match on the final extracted numerical answer after normalizing for commas and trailing decimal zeros. We use the evaluation scripts from the `openai/simple-evals` repository (7), adapted for local model inference using vLLM. Prompts are formatted in the target language with language-specific instructions that request reasoning followed by a numeric answer in a specified format. Models are sampled with greedy decoding (temperature=0.0) and a maximum generation length of 2,048 tokens. Reported results correspond to the mean accuracy across the 11 languages.

We make the following observations:

- Google's gemma-3-27b-it significantly outperformed the others, scoring around 88% while OpenAI's gpt-oss-20b got second place with around 77%.

- For the first time, Qwen3-30B-A3B-Instruct-2507 performed the worst in a benchmark, scoring around 70%.

Benchmark 3: IF Bench

The Instruction Following benchmark (IFBench) from AllenAI assesses instruction-following across diverse tasks like counting, formatting, and text manipulation (13). The models are evaluated on how well they follow constraints in their responses, such as “Include keyword beneath in the 23rd sentence, as the 34th word of that sentence,” “Use at least 3 different coordinating conjunctions in the response,” or “The second word in your response and the second to last word in your response should be the word vibrant.”

Experimental setup. We use the 294-question single-turn dataset with 5 repetitions per question, evaluated via pass@1 scoring (1). Responses are scored using the repository’s official evaluation code in loose mode, which tolerates formatting variations and extraneous text by testing multiple output formats (removing leading/trailing lines, asterisks, etc.). Results reflect average prompt-level accuracy across all questions and runs. In our experiments, models are sampled with temperature 0.7, top-p 0.95, and a 4096 token limit.

We make the following observations:

- The model that scored the lowest in the prior benchmarks, gpt-oss-20b, significantly outperformed the others in this evaluation, scoring about 55%.
- The other four models were close in performance, scoring between 32% and 40%.

Benchmark 2: GPQA Diamond

The GPQA (Graduate-Level Google-Proof Q&A) Diamond dataset consists of highly challenging multiple-choice problems in biology, physics, and chemistry, requiring scientific knowledge and reasoning abilities to answer. Each question is authored by subject-matter experts and is intentionally formulated to be difficult for non-experts to solve, even with access to online resources. The most difficult “Diamond” split includes 198 questions for which both expert annotators selected the correct answer, while most non-experts did not.

Experimental setup. We evaluate the models following the guidelines from the benchmark authors (14) and using the OpenAI simple-eval library (7). Every question is presented 10 times, each time with the answer choices in a different randomized order. The models are allowed a maximum output length of 2028 tokens.

We make the following observations:

- The top four models all achieved accuracies within 4.5 percentage points of one another.
- The lowest-performing model, gpt-oss-20b, again scored noticeably lower than the other models.

Benchmark 1: MMLU-Pro

MMLU-Pro (17) is an enhanced benchmark for evaluating language models, building on the original MMLU dataset with significantly harder, reasoning-intensive questions and 10 answer options instead of 4. It contains over 12,000 curated questions from academic sources spanning 14 fields, including Biology, Computer Science, Mathematics, Physics, and Law. Experiments demonstrate MMLU-Pro substantially increases difficult with model accuracies dropping 16–33% compared to standard MMLU. Notably, Chain-of-Thought prompting yields greater improvements on MMLU-Pro than on the original benchmark, indicating the dataset demands deeper, more structured reasoning.

We observed the following from the performance of the Delphi entrant models.

- Qwen3-30B-A3B-Instruct-2507 performed the best, scoring nearly 75%.
- The first place model scored less than 10% ahead of the fourth place model.
- The last place model, gpt-oss-20b, scored noticeably lower than the other models.

We evaluate the models following the guidelines from the benchmark authors (on `TIGER-Lab/MMLU-Pro repository` (8)). The models are allowed a maximum output length of 2048 tokens.

References

[1] allenai/IFBench: A new, challenging benchmark for precise instruction following. https://github.com/allenai/IFBench.

[2] Artificial Analysis: LiveCodeBench Leaderboard. https://artificialanalysis.ai/evaluations/livecodebenchl.

[3] cais/hle: a multi-modal benchmark at the frontier of human knowledge, designed to be the final closed-ended academic benchmark of its kind with broad subject coverage. https://huggingface.co/datasets/cais/hle.

[4] evalplus/evalplus: Rigourous evaluation of LLM-synthesized code. https://github.com/evalplus/evalplus/tree/master.

[5] LiveCodeBench/LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code. https://github.com/LiveCodeBench/LiveCodeBench.

[6] OpenAI: Introducing GPT-5. https://openai.com/index/introducing-gpt-5/.

[7] OpenAI simple-eval: a lightweight library for evaluating language models. https://github.com/openai/simple-evals/tree/main.

[8] TIGER-AI-Lab/MMLU-Pro: A More Robust and Challenging Multi-Task Language Understanding Benchmark. https://github.com/TIGER-AI-Lab/MMLU-Pro, 2024.

[9] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[10] Xueying Du, Mingwei Liu, Kaixin Wang, Hanlin Wang, Junwei Liu, Yixuan Chen, Jiayi Feng, Chaofeng Sha, Xin Peng, and Yiling Lou. Evaluating large language models in class-level code generation. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, ICSE '24, New York, NY, USA, 2024. Association for Computing Machinery.

[11] Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[12] Alex Gu Wen-Ding Li Fanjia Yan Tianjun Zhang Sida Wang Armando Solar-Lezama Koushik Sen Ion Stoica Naman Jain, King Han. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint*, 2024.

[13] Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. Generalizing verifiable instruction following, 2025.

[14] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

[15] Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. Language models are multilingual chain-of-thought reasoners, 2022.

[16] 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, 2024.

[17] Y. Wang, X. Ma, G. Zhang, Y. Ni, A. Chandra, S. Guo, W. Ren, A. Arulraj, X. He, Z. Jiang, T. Li, M. Ku, K. Wang, A. Zhuang, R. Fan, X. Yue, and W. Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024.