Muse Spark Eval Methodology

We evaluate Muse Spark models over a diverse set of benchmarks including reasoning, multimodal, coding, tool use, as well as health knowledge.

Overall Methodology

- **We compare our model to a basket of other industry models:** Gemini 3.1 Pro, GPT 5.4, Claude Opus 4.6, and Grok 4.2. For those, we report the most favorable result between self-reported scores or our internal reproductions from the model's API. We use the highest reasoning effort for both our model and other models, specifically high reasoning effort for Gemini, xhigh reasoning effort for GPT, max reasoning effort for Claude, and the reasoning variant for Grok. For coding and agentic benchmarks, we report self-reported results and only run internal evals when self-reported results are not available. For all models, we run a single attempt per problem unless otherwise specified.
- **Multimodal evals with Python:** we perform evaluation with a Python interpreter as a server-side tool. When the model issues a Python snippet, we execute the code in a sandbox Python environment and return the output to the model, the output can be text or images. Search and internet access are prohibited during python execution. We directly call API with code execution / tool enabled following the official instructions, specifically Code Interpreter (for GPT) and Code Execution (for Gemini).
- **Agentic evaluations for third-party models:** the results represent our best-effort evaluations of third-party models, obtained using the same evaluation framework as for our internal models to ensure consistency. We note that our evaluation setup (e.g. agent tools and system prompts) may not be specifically tuned for proprietary third-party models. Therefore, the results may not reflect these models' best performance when used in environments tailored to their specific strengths.

Per-benchmark details

- **Humanity's Last Exam (HLE):** We use the full 2,500-question dataset and GPT o3-mini as the LLM-as-judge model, following the official HLE evaluation protocol. For results with tools, we allow bash and browser tools with blocklist on huggingface.co, kaggle.com, paperswithcode.com, lmsys.org.

- **GPQA Diamond:** results are averaged over 4 runs to reduce variance.

- **ARCAGI 2:** results are based on the public set with 120 prompts. We report on pass@2 to follow the setup of the arc-agi2 competition on kaggle.

- **HealthBench Hard:** This is a subset of OpenAI's HealthBench benchmark, containing 1000 prompts. We used the same implementation as in the OpenAI's official simple-evals repo, with GPT-4.1-genai as the LLM-as-judge model.

- **MedXpertQA Text/Multimodal:** The text variant has 2,450 prompts and spans medical specialties with 10 answer choices (A–J); The multimodal variant contains 2,000 multimodal medical questions with clinical images (X-rays, histology, dermatology, etc.) and 5 answer choices (A–E). For grading, we use gpt-oss-120b to parse the predicted answer letter from free-form text.

- **CharXiv Reasoning**: We use 1,000 chart reasoning questions from the CharXiv validation set. Grading uses gpt-oss-120b as the LLM-as-judge model, which evaluates semantic and mathematical equivalence between the model's response and the ground-truth answer.

- **MMMU Pro:** We evaluate on MMMU Pro standard set (10 options) and models are prompted to output their answer in \boxed{ANSWER} format. Grading uses a rule-based method that extracts the last matching answer pattern.

- **ERQA:** We use 400 embodied reasoning questions with multiple-choice answers (A–D) testing spatial reasoning in robotic/embodied contexts. Models are instructed to produce step-by-step solutions followed by a boxed single letter, and responses are graded by exact match of the selected letter against the ground truth answer.

- **SimpleVQA:** We use 2,024 multimodal visual question-answering questions covering both English and Chinese subsets. We use the same implementation as in the official SimpleVQA github repo, where grading uses GPT 4o as LLM-as-judge model that classifies answers as Correct, Incorrect, or Unattempted. We report accuracy (correct/total) as the primary metric.

- **Screenspot Pro:** We use 1,581 examples from the official benchmark, and instruct the model to output the normalized coordinate of UI elements. We report the accuracy for the predicted point in the ground truth box. During our benchmarking process, we suspect Grok 4.2 is likely not using the normalized coordinate as

native representation, to maintain the integrity and fairness of our comparisons, we decided to omit the result until we can ensure it accurately reflects Grok 4.2's true performance. Additionally, we used (y, x) coordinate convention for Gemini according to the official API doc. Screenspot Pro involves high resolution image understanding and tools frequently bring gains. For the with-python version, we attempt to use the variant of tool use that gives each model the most advantage. For our model and Gemini, we used a cropping tool; and for GPT we enabled a python code interpreter. We resize large images to fit within API limits of each model.

- **Zerobench:** We use the 100 main questions dataset, and grade the response using gpt-oss-120b as the LLM-as-judge model that compares the model's free-form answer against a reference answer. Following the zerobench leaderboard, we report pass@5 as the primary metric. We resize large images to fit within API limits of each model.

- **SWE-Bench Verified and SWE-Bench Pro (public set):** Our agent scaffolding uses a single-attempt setup consisting of a bash tool and a file operation tool for viewing and editing files. Evaluation is performed by running model-predicted code solutions using their official evaluation scripts and docker images. Results are averaged over 15 attempts per problem for SWE-Bench Verified (Grok 4.2 was evaluated with 10 samples per prompt) and 4 attempts for SWE-Bench Pro. For SWE-Bench Verified, we follow the evaluation methodology of Gemini 3.1 Pro and fixed bugs in the tests for three problems (`astropy__astropy-7606`, `sphinx-doc__sphinx-8595` and `sphinx-doc__sphinx-9711`) for which even running the ground-truth solutions won't pass. Additionally, we also fixed the tests in another two problems (`sphinx-doc__sphinx-8269` and `sphinx-doc__sphinx-8475`) that also failed to run on our infrastructure due to internet access issues.

- **LiveCodeBench Pro** is a competitive programming benchmark sourced from Codeforces problems. We evaluate the 2025Q2 subset (Apr–Jun 2025) in C++, using the set of prompts, tests, and checkers aligned with the LiveCodeBench Pro release. 2025Q2 is the most recent split with complete hidden tests provided. Models generate code solutions that we execute against the full test suites with checkers using our internal Code Execution Service, which is configured to match the official Codeforces contest environment. We report pass@1 averaging over 4 attempts per problem to reduce variance. We extract pass@1 for Gemini 3.1 pro

directly from the LiveCodeBench Pro 2025Q2 leaderboard as the self-reported result.

- **Terminal-Bench 2.0**: We performed evaluation with a bash-tool-only agent scaffold, where the agent follows the reasoning and acting framework to resolve tasks with bash commands. Each task runs in a container with 6 CPU cores and 8GB RAM. Results are averaged over 15 attempts. We used the official terminal-bench-2 repository and fixed the test script of one problem in order to work with our infrastructure. Specifically, for task `sam-cell-seg`, the test script uses `uvx` which cannot specify +cpu local version tags for PyTorch wheels, so on CPU-only containers it fails to resolve the correct package version. We fixed it by switching back to `pip install` with explicit `torch==2.5.1+cpu` and `torchvision==0.20.1+cpu` pinning.

- **DeepSearchQA** is an agentic browsing evaluation where models use browser tools (search, open, find) to answer questions where the answer is a list of items. Grading uses gpt-oss-120B as the judge to extract the model's predicted answer set, semantically matches predictions to targets, and computes an F1 score. We used the same search engine we selected across all models.

- **τ-BenchTelecom** results are sourced from the Artificial Analysis τ^2-Bench Telecom Leaderboard.

- **GDPval-AA** results are sourced from the Artificial Analysis GDPval-AA Leaderboard.

	Benchmark	Muse Spark Thinking	Opus 4.6 Max	Gemini 3.1 Pro High	GPT-5.4 Xhigh	Grok 4.2 Reasoning
MULTIMODAL	CharXiv Reasoning Figure Understanding	86.4 w/ Python: 88.9	65.3 w/ Python: 78.9	80.2 w/ Python: 80.7	82.8 w/ Python: 85.9	60.9
	MMMU Pro Multimodal Understanding	80.4 w/ Python: 81.3	77.4	83.9 w/ Python: 84.1	81.2 w/ Python: 82.8	75.2
	ERQA Embodied Reasoning	64.7 w/ Python: 67.0	51.6	69.4 w/ Python: 72.4	65.4 w/ Python: 73.5	54.1
	SimpleVQA Visual Factuality	71.3 w/ Python: 76.1	62.2	72.4 w/ Python: 73.0	61.1 w/ Python: 61.8	57.4
	ScreenSpot Pro Screenshot Localization	72.2 w/ Python: 84.1	57.7 w/ Python: 83.1	61.0 w/ Python: 84.4	39.0 w/ Python: 85.4	—
	ZeroBench Multi-Step Visual Reasoning (pass@5)	15.0 w/ Python: 33.0	11.0	19.0 w/ Python: 29.0	23.0 w/ Python: 41.0	9.0
TEXT/REASONING	Humanity's Last Exam Multidisciplinary Reasoning (No Tools)	42.8	40.0	45.4 Self-Reported: 44.4	43.9 Self-Reported: 39.8	31.6
	Humanity's Last Exam Multidisciplinary Reasoning (With Tools)	50.4	53.1	51.4	52.1	—
	ARC AGI 2 Abstract Reasoning Puzzles (Public)	42.5	63.3	76.5	76.1	53.3
	GPQA Diamond PhD Level Reasoning	89.5	92.7 Self-Reported: 91.3	94.3	92.8	88.5
	LiveCodeBench Pro Competitive Coding	80.0	70.7	82.9 Self-Reported: 78.2	87.5	74.2
HEALTH	HealthBench Hard Open-Ended Health Queries	42.8	14.8	20.6	40.1	20.3
	MedXpertQA (Text) Medical Multiple Choice	52.6	52.1	71.5	59.6	50.2
	MedXpertQA (MM) Medical Multiple Choice	78.4	64.8	81.3	77.1	65.8
AGENTIC	DeepSearchQA Agentic Search	74.8	73.7	69.7	73.6	62.8
	SWE-Bench Verified Agentic Coding	77.4	80.8	80.6	—	76.7*
	SWE-Bench Pro Diverse Agentic Coding	52.4	53.4	54.2	57.7	51.8*
	Terminal-Bench 2.0 Agentic Terminal Coding	59.0	65.4	68.5	75.1	47.1*
	τ^2-Bench Telecom Agentic Tool Use (Artificial Analysis)	91.5	92.1	95.6	91.5	96.5
	GDPval-AA Elo Office Tasks (Artificial Analysis)	1444	1606	1320	1672	1055

Contemplating Mode

Contemplating mode leveraged a novel multi-round test-time scaling scaffold (solution generation, iterative self-refinement, aggregation) to improve model performance. All the results are evaluated on the final answer output by the scaffold.

- Humanity's Last Exam (HLE): same as above
- LiveCodeBench Pro: same as above
- IPhO: International Physics Olympiad 2025 (theory) comprises all 3 theory problems from the official 2025 IPhO competition: results are based on (blinded) human evaluation with guidelines based on the official competition scoring, and validated by domain experts. Results are averaged over three generations per problem.
- FrontierScience Research: for Gemini 3.1 DeepThink, we query via the DeepThink web interface and evaluate with 1 sample per prompt. The Contemplating mode number is averaged over 8 samples per prompt and uses identical judge prompt+model to the third-party eval.

Benchmark	Muse Spark Contemplating	Gemini 3.1 Deep Think	GPT 5.4 Pro
Humanity's Last Exam Multidisciplinary Reasoning (No Tools)	50.2	48.4	43.9 Self-Reported: 42.7
Humanity's Last Exam Multidisciplinary Reasoning (With Tools)	58.4	53.4	58.7
IPhO 2025 (Theory) Physics Olympiad	82.6	87.7	93.5
FrontierScience Research Scientific Research	38.3	23.3*	36.7