# Muse Spark 1.3

## Evaluation Methodology

We evaluate Muse Spark 1.3 on benchmarks spanning professional and computer-use agents, web research and automation, software engineering, instruction following, and long-context retrieval.

### Overall Methodology

- **Models and configurations.** Muse Spark 1.3 results are generated through the Meta Model API. Where comparable results are available, we compare with Muse Spark 1.2, Claude Opus 5, and GPT-5.6 Sol. We use xhigh reasoning effort for Muse Spark 1.3 and Muse Spark 1.2, and max for Claude Opus 5 and GPT-5.6 Sol. Agentic coding uses the named agent product or fixed harness; other agentic evaluations use the benchmark provider's harness or a common internal framework. Tools are limited to those required by each benchmark, and coding environments do not have external internet access.
- **Evaluation and score selection.** We preserve each benchmark's tasks, inputs, grading, and primary metric as closely as possible. For each model, we report the highest comparable primary-metric value available from our evaluation, the official leaderboard, or the model provider's self-reported results. Benchmark names link to official leaderboards where available. In our evaluations, ungradable responses and model refusals receive zero credit and remain in the denominator rather than being dropped. Material departures from the official setup are noted below.
- **Third-party models.** Our runs use common settings where practical but are best-effort: prompts, tools, and runtime may not be tuned for proprietary models and may not reflect their best provider-optimized performance.

### Per-benchmark details

#### Professional work

- **GDPVal-AA v2.** GDPVal-AA v2 evaluates agents on 220 real-world professional tasks from OpenAI's GDPval benchmark, described in the paper. The tasks span 44 occupations and nine major U.S. industries and require deliverables such as documents, spreadsheets, slide decks, diagrams, and reports. Models operate in Artificial Analysis's Stirrup agentic harness with shell access and web browsing. An LLM judge compares anonymized deliverables head-to-head, and the primary metric is the Elo rating derived from these blind pairwise comparisons, with the

human baseline anchored at 1,000. Artificial Analysis publishes results on the linked official leaderboard.

- **JobBench.** JobBench is a benchmark of 65 professional tasks across 35 white-collar occupations. Each task asks the agent to produce a practical work product and is scored against task-specific rubrics. We use the official task set and a file-aware rubric grader. The primary metric is the mean rubric score across tasks. Results sourced from JobBench follow the official evaluation code with the OpenCode harness; the benchmark setup is described in the paper.

### Computer use

- **OSWorld 2.0.** OSWorld 2.0 is a long-horizon computer-use benchmark of 108 real-world workflows on a full Ubuntu desktop VM. Each task is an end-to-end workflow spanning multiple applications, with a coherent stateful user profile, authentic input artifacts, and a dynamic environment. The agent is given a GUI computer-control tool and interacts by taking screenshots and emitting actions such as clicking, typing, scrolling, and key presses. We run OSWorld 2.0 in a common internal evaluation framework using the validated benchmark release available for each model, while preserving the corresponding official task environments and grading logic. The paper and official implementation describe the benchmark setup. Grading is execution-based through each task's programmatic checker. The primary metric is the mean per-task partial score; strict binary completion is reported as a secondary metric. All results on OSWorld 2.0 were evaluated using version 08.08, except Muse Spark 1.2, which was evaluated on version 06.24.

### Web research and automation

- **DeepSearchQA.** DeepSearchQA contains 900 agentic browsing questions from the official dataset whose answers are lists of items. Models use browser tools such as search, open, and find to research each question. Grading extracts the model's predicted answer set and semantically matches predictions to the targets. We use the same search backend and browser harness across all models. We use F1-score as the primary metric.
- **AutomationBench.** AutomationBench, from Zapier, evaluates agents on 600 realistic business workflows in a simulated application environment, as described in the paper. The agent completes each workflow through the benchmark's automation tools. Deterministic end-state assertions inspect the resulting application state, with no language-model judge. We use the public v3 task set; Zapier publishes the official public leaderboard in the benchmark repository. The primary metric is pass@1, the percentage of workflows completed successfully.

### Software engineering and code understanding

- **DeepSWE v1.1.** DeepSWE is a long-horizon software-engineering benchmark with 113 tasks across 91 repositories and five languages: TypeScript, Go, Python, JavaScript, and Rust. The tasks are written from scratch and graded with handwritten functional and regression tests. We run

Muse Spark 1.3 max with a mini-swe agent and obtained the benchmarks for all other models on the official leaderboard on Datacurve. The primary metric is task pass rate: the percentage of tasks whose functional and regression tests both pass. The benchmark is described in the paper.

- **SWE-Marathon.** SWE-Marathon v1.1 contains 20 project-scale tasks spanning library rewrites and reproductions, product clones, ML engineering, and algorithmic or optimization work. We run the official task set in isolated cloud sandboxes, including GPU environments for the five ML tasks, and use each task's verifier to grade the final result. The primary metric is mean pass@1: partial scores do not count as task success. The benchmark is described in the paper.
- **SWE-Atlas Codebase QnA.** SWE-Atlas Codebase QnA comprises 124 tasks evaluating deep code comprehension—the upstream capability that precedes a code change. The benchmark spans 11 production repositories across Go, Python, C, and TypeScript. We use the public QnA split with the mini-swe-agent harness and task-specific rubric grading, following Scale AI's leaderboard protocol and official implementation. The primary metric is mean pass@1 across tasks.
- **Terminal-Bench 2.1.** Terminal-Bench 2.1 contains 89 tasks completed in terminal environments. We run each task with the coding agent named in the result using our internal agent evaluation framework and an isolated cloud sandbox. The official executable verifier grades the final container state. The primary metric is mean pass@1 across tasks. The Terminal-Bench paper describes the benchmark framework. GPT 5.6 Sol benchmark was obtained directory from OpenAI.

### Long-context retrieval

- **MRCR v2 (256K–512K and 512K–1M).** MRCR v2 (Multi-Round Co-reference Resolution) is a long-context retrieval benchmark. We use OpenAI's MRCR v2 data, re-binned by o200k_base token count, and evaluate 100 examples in each of the 256K–512K and 512K–1M bands using the 8-needle variant. Each task requires retrieving the target turn from among 8 needles distributed across the full context. It is pure retrieval-from-context with no agent tools: the model must reproduce a target string via retrieval and reasoning prefixed by a per-sample random hash. Grading is rule-based, using a sequence-matcher ratio between the response and the target, and we report the mean sequence-matcher ratio as the primary metric.

### Meta internal instruction-following index

- **IF Index.** IF Index is an internal composite evaluation of whether a model follows the explicit constraints and process requirements in agentic tasks. It aggregates multiple internal evaluations rather than a single fixed task set, so no single task count applies. It covers adherence to tool-use constraints, fine-grained rubrics, compound constraints, long policy requirements, and explicit instructions embedded in structured workflows.

| | Benchmark | Muse Spark 1.3 (max)<br>Meta | Muse Spark 1.3 (xhigh)<br>Meta | Muse Spark 1.2 (xhigh)<br>Meta | GPT 5.6 Sol (max)<br>OpenAI | Opus 5 (max)<br>Anthropic |
|---|---|---|---|---|---|---|
| Agent | **GDPVal-AA v2**<br>Knowledge work | 1754 | 1709 | 1615 | 1710 | **1824** |
| | **JobBench**<br>Professional tool use | 64.9 | 61.2 | 61.6 | 45.4 | **65.7** |
| | **OSWorld 2.0**<br>Agentic computer use | 66.9 | 57.2 | 47.6 | 62.7 | **68.3** |
| | **DeepSearchQA**<br>Agentic browsing | 89.4 | 89.4 | 85.9 | **93.0** | 90.4 |
| | **Agentic IF Index (Internal)**<br>Instruction following | 57.8 | 55.7 | 46.2 | **60.5** | 59.1 |
| | **AutomationBench**<br>E2E business workflows | 49.4 | 43.8 | 38.2 | 46.7 | **50.3** |
| Long Context | **MRCR 256K-512K**<br>Long context | **98.5** | 97.6 | 66.3 | 91.5 | - |
| | **MRCR 512K-1M**<br>Long context | **98.1** | 93.1 | 55.5 | 73.8 | - |
| Coding | **DeepSWE v1.1**<br>Long-horizon agentic coding | **75.4** | - | 55.0 | 73.0 | 74.0 |
| | **SWEAtlas CodeBase QnA**<br>Codebase understanding | **59.4** | 54.0 | 46.2 | 53.5 | 52.7 |
| | **Terminal-Bench 2.1**<br>Agentic terminal coding | 88.8 | **89.2** | 82.9 | 88.8 | 86.7 |