Muse Glimmer Evaluation Methodology

We evaluate Muse Glimmer over a diverse set of benchmarks including general agentic, agentic coding, multimodal / computer use, safety, reasoning, and general capability benchmarks.

Overall Methodology

- **We compare Muse Glimmer to other leading open-weight models of similar size and architecture.** For models other than Muse Glimmer, we report the most favorable result between self-reported scores or our internal reproductions from the locally serving model. We report numbers from Artificial Analysis if they are available for all models.
- We use the following thinking and sampling configuration settings:
    - For Muse Glimmer, we use high reasoning strength, and temperature=1.0/top_p=0.95/top_k=64 across all benchmarks.
    - For Gemma4-31B, we follow the recommended sampling configuration and we use thinking mode and temperature=1.0/top_p=0.95/top_k=64 across all benchmarks
    - For Qwen3.6-27B, we follow the recommended sampling configuration and we use thinking mode and temperature=1.0/top_p=0.95/top_k=20 across all benchmarks, except for GAIA2 and WildClawBench, where we use temperature=0.6/top_p=0.95/top_k=20; this is to align with usage of temperature=0.6 for QwenClawBench.
- **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.

Muse Glimmer per-benchmark details

General Agentic

- **MCP-Atlas**: MCPAtlas is a multi-turn agentic tool-use evaluation where the model uses 20+ MCP tool servers (such as GitHub, Twelve Data, Notion) to answer queries. Grading happens using an LLM judge (Gemini 2.5 Pro) that assesses whether the final model answer fulfills the ground-truth claims. The primary metric is mean pass rate, where a task successfully passes if it has a score higher than the 0.75 threshold. The benchmark consists of the 500 public tasks and results are averaged over 4 runs to reduce variance. We report internal run results for all 3 models.
- **DeepSearchQA**: DeepSearchQA is an agentic browsing evaluation where the model answers 900 hard research questions by acting as an autonomous web-browsing agent.

It is given a single browsing tool with three functions: search, open and find. It works in a multi-step loop: it reasons, calls a function, receives the result as a new message, and repeats until it produces a final answer. 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. Results are averaged over 4 runs to reduce variance. We report internal run results for all 3 models.

- **τ^3-bench Banking** is a knowledge-retrieval-based customer service evaluation extending τ-bench to a fintech banking domain. The model acts as a customer-service assistant that must retrieve relevant information from unstructured knowledge documents (policy guides, FAQ pages, compliance procedures) via configurable RAG pipelines, document search, and agentic shell-based search to resolve customer queries. The model interacts against a simulated user, with task success determined by satisfying per-task ground-truth assertions about the retrieved information and generated responses. Results are sourced from Artificial Analysis following the documented benchmarking methodology.
- **WildClawBench**: WildClawBench is an open-source benchmark for evaluating agentic models on real-world, long-horizon tasks. Its 60 tasks span six categories: productivity flow, code intelligence, social interaction, search and retrieval, creative synthesis, and safety alignment. Collectively, these tasks evaluate multi-step planning, coordinated tool use, multimodal understanding, information retrieval, coding, cross-modal content creation, and safe behavior. Each task runs in an isolated Docker container under the OpenClaw harness, with access to tools such as a browser, shell, file system, email, and calendar. Task-specific grading uses deterministic verification criteria, GPT-5.4 as an LLM judge, or a combination of both, depending on the task. Each grader produces an overall score between 0 and 1 by aggregating its task-specific criteria. We report the unweighted mean of the 60 task scores, averaged over three independent runs per task to reduce variance. We report internal run results for all 3 models.
- **GDPval-AA v2:** GDPval-AA v2 assesses models on economically valuable knowledge work spanning 44 occupations across key sectors contributing to US GDP. The benchmark contains 220 tasks developed by OpenAI in collaboration with industry professionals to reflect real-world complexity. Each task requires the model to produce one or more deliverable files, run as an agent in AA's open-source Stirrup harness with an E2B code-execution sandbox, a 250-turn budget and access to a range of available tools. Results are sourced from Artificial Analysis following the documented benchmarking methodology.
- **Gaia2:** Gaia2 is an open-source, text-only agentic tool-use benchmark that evaluates general-purpose assistants on multi-turn tasks inside simulated, stateful "universes" of everyday applications, such as email, calendar, messaging, contacts, shopping, and more. Each scenario runs in an isolated container hosting a self-contained environment: the agent fulfills user requests by issuing tool calls against the environment's app APIs, while an event daemon injects time-based and environment-driven events mid-episode (e.g., incoming messages, state changes, scheduled triggers) that the agent must notice and adapt to. We use the public Github repository for our evaluation, which contains 800

tasks across five capability splits (execution, search, ambiguity, adaptability, and time). We use the OpenClaw harness for our experiments. We measure the mean per-sample success rate over three independent runs per task, matching the public leaderboard convention. Grading is LLM-judged: an in-container judge compares the agent's realized tool actions and end state against each scenario's oracle events to produce a per-task success score; we use gpt-oss-120b as the reference judge. We report internal run results for all 3 models.

- **SkillsBench:** SkillsBench is an open-source benchmark that measures how effectively an agent can leverage "skills", modular folders of instructions and helper scripts, to complete specialized knowledge work tasks. We use the 86-task set, spanning a broad range of software and data engineering domains. Each task runs in an isolated container preconfigured from the task's Docker image and init commands with the relevant skill folders mounted into the workspace. The harness exposes a bash/terminal tool plus image-view and browser tools. Grading is deterministic and pytest-based: after the agent finishes, a hidden test suite is uploaded into the same container and executed, and the reward is the weighted fraction of tests passed. We report the mean per-task reward across four attempts. We report internal run results for all 3 models.
- **OSWorld-Verified:** OSWorld-Verified is a multimodal benchmark for computer-use agents that evaluates the ability to operate a full Ubuntu desktop VM via GUI in realistic, cross-app workflows. We evaluate on the 361-task split, which excludes 8 Google Drive tasks. Agents are restricted to GUI-based computer control with no direct shell access: they observe the environment through screenshots and act via GUI actions such as click, type, scroll, and key press. Grading is execution-based, where each task defines programmatic getters and checkers that inspect the final VM state and return a 0-1 reward. Models receive 1920 × 1080 screenshots and predict click locations in a normalized coordinate space, with the ability to batch multiple GUI actions per step. Each episode is capped at 200 steps with a screenshot history truncation mechanism enabled that retains a maximum of 19 most recent screenshots. For Qwen3.6-27B and our model, we adopt the Claude computer-use action space (version computer_20251124) with a normalized 0-1000 coordinate space plus a separate stop action. For Gemma4-31B, we use the Gemini 2.5 Flash computer-use interface, including click_at, type_text_at, and stop, with relative coordinates in [0,999]. Following the official OSWorld protocol, we report mean per-task reward. The final score is averaged over four attempts per task to reduce variance. We report internal run results for all 3 models.

Agentic Coding

- **SWE-Bench Verified and SWE-Bench Pro** feature 500 and 731 diverse and complex agentic coding tasks, respectively. Our agent scaffolding consists of a bash tool and a file operation tool for viewing, creating and editing files. We report average pass rate across tasks and we average results across 4 runs to reduce variance. For SWE-Bench Verified, we report Qwen3.6-27B's self-reported score. For SWE-Bench Pro, we do not use the Qwen3.6-27B's self-reported score which was measured on a refined version of

the benchmark as we evaluated on the original task set. We report internal run results for the remaining cases.

- **Terminal-Bench 2.1**: Terminal-Bench 2.1 (Stanford × Laude Institute) evaluates an agent on real-world tasks in a terminal environment. It contains 89 curated tasks across software engineering, system administration, data processing, model training, and security. The Terminus 2 agent harness is used in an E2B sandbox environment for the evaluation. Results are sourced from Artificial Analysis following the documented benchmarking methodology.
- **SciCode:** SciCode contains scientist-curated laboratory problems across 16 scientific disciplines, developed by domain experts with the reported evaluation using the 288 test set subproblems. The benchmark tests Python programming to solve scientific computing tasks: models must generate code that passes every unit test for a sub-problem to be credited. We prompt with the scientist-annotated background information included, and report sub-problem-level scoring rather than whole-problem scoring. No tool use is permitted. Results are sourced from Artificial Analysis following the documented benchmarking methodology.

Multimodal

- **CharXiv Reasoning:** We use 1,000 chart reasoning questions from the CharXiv validation set. Grading uses gpt-oss-120b as an LLM judge, which evaluates semantic and mathematical equivalence between the model's response and the ground-truth answer. These multimodal questions are paired with diverse, complex scientific charts sourced directly from academic papers across multiple disciplines. The benchmark tests a model's ability to go beyond simple data extraction to perform complex visual synthesis, multi-step mathematical calculations, pattern recognition, and trend comparisons. We report the average task success rate across four attempts. We report internal run results for Gemma4-31B and our model, and a self-reported number for Qwen3.6-27B.
- **ScreenSpot-Pro:** ScreenSpot-Pro is an open-source multimodal GUI-grounding benchmark that evaluates whether a model can locate a requested interface element in a static screenshot. We evaluate the full1,581-example ScreenSpot-Pro set, which emphasizes high-resolution professional applications across Windows, macOS, and Linux. Each example provides a screenshot and a natural-language description of the target element. Models utilize an iterative Python cropping tool. The model first predicts a point on the full screenshot in normalized [0, 1000] coordinates. We then render a grid-overlaid zoom around that point and map the model’s crop-local prediction back to the original image. This loop repeats until the prediction converges, supporting up to 10 verification rounds and allowing up to three restarts from the original picture if the model considers that the target is absent from a crop. Grading is deterministic: a prediction receives reward 1 if the final point lies inside the annotated target bounding box and 0 otherwise. We report mean point-in-box accuracy (pass@1), averaged over four independently sampled predictions per example to reduce variance. We report internal run results for all 3 models.

- **OmniDocBench v1.5:** We use the v1.5 version of the benchmark containing 1355 multimodal prompts. This benchmark evaluates a model's ability to accurately parse, extract, and reason over highly complex, multi-format documents. The inputs consist of diverse, real-world document pages, such as academic papers, textbooks, financial reports, slides, and handwritten notes, that challenge the model to simultaneously process dense text, complex multi-column layouts, tables, and embedded mathematical formulas. We report average scores across all tasks using an internal implementation of the original scoring protocol, that is modified to replace the three-component scoring formula (text, tables, formulas) with a two-component weighted average that folds formulas into text groups (graded by edit distance) rather than evaluating them separately with the Character-and-Delimiter Match metric. Our implementation uses a simpler Hungarian algorithm for bipartite element matching instead of the official v1.6 MGAM (Multi-Granularity Adaptive Matching) which adaptively searches for optimal prediction segmentation granularity while keeping ground truth fixed. Results are averaged over 4 runs to reduce variance. We report internal run results for all 3 models.
- **MMMU-Pro:** MMMU-Pro is a multimodal reasoning benchmark consisting of 1730 multiple choice questions. The benchmark presents college-level problems spanning 30 academic subjects across STEM, humanities, and social sciences, specifically filtered to remove any questions answerable by text-only models. The questions require expert-level reasoning and are paired with highly complex, heterogeneous image types, including diagrams, chemical structures, medical scans, and charts, where the answer choice space has been significantly expanded to minimize the success of random guessing. Results are sourced from Artificial Analysis following the documented benchmarking methodology.

Safety

- **CIMemories:** CIMemories is a benchmark measuring how much models reveal personal sensitive information in inappropriate contexts. CIMemories uses synthetic user profiles with over 100 attributes per user (such as finance, health, relationships attributes) paired with diverse task contexts representing canonical social interactions (e.g., communicating with doctors) resulting in a total of 2500 scenarios. Each attribute may be essential for some tasks but inappropriate for others, giving rise to two primary metrics, violation rate (the extent to which inappropriate attributes are revealed) and coverage rate (the extent to which necessary attributes are shared). We use Claude 4.6 Sonnet as an LLM judge to determine sharing of attributes by the model and we average results over 4 runs to control variance. We report internal run results for all 3 models.
- **Siren AgentDojo:** Siren AgentDojo is a public benchmark designed to evaluate how models handle standard tool-use scenarios when exposed to prompt injection attacks. The evaluation combines 97 benign tasks spread across four distinct environments (banking, travel booking, workspace management, online store) with 35 malicious tasks yielding a total of 949 prompt injection attempt scenarios. Each of these equips the agent with domain-specific tools, such as transaction APIs, calendar access, and file operations. The benchmark specifically tests whether agents can be manipulated by

stealthy attacks into misusing these tools for malicious actions like data exfiltration or unauthorized financial transactions. The LLM-powered attacker (we use Claude Opus 4.6) iteratively refines its attack until it succeeds or a maximum of 6 iterations is reached. We use deterministic rule-based grading to determine attack success and benign task completion. We report two scores here: Attack Success Rate (ASR), which is the success rate of the malicious task injected by the attacker and utility, which is the success rate of the benign task instructed by the user in the presence of a malicious injection. Higher utility and lower ASR indicate that the agent is able to complete more user requests, while being less impacted by the injected prompts. We report internal run results for all 3 models.

Chem/Bio Domain

- **MBCT, HPCT, and VCT:** The Molecular Biology Capabilities Test (MBCT), Virology Capabilities Test, multi-modal (VCT), and Human Pathogens Capabilities Test (HPCT) are part of a suite of evaluations developed by SecureBio and the Center for AI Safety. These evaluations are designed to assess practical troubleshooting across a range of molecular biology tasks (MBCT), wet lab virology experiments (VCT), and practical knowledge about working with human pathogens considered high-priority by biosecurity experts (HPCT). Our performance metric is accuracy.
- **WMDP (Bio, Chem):** The Weapons of Mass Destruction Proxy (WMDP) evaluation assesses dual-use conceptual knowledge in harmful domains. Our performance metric is accuracy.
- **LAB-Bench:** The Language Agent Biology Benchmark (LAB-Bench) is an evaluation suite designed to assess AI capabilities on practical biology research tasks essential for scientific research including protocol planning and data analysis.The ProtocolQA task assesses the ability to debug practical wet-lab protocols. Our performance metric is accuracy.

Cyber Domain

- **CyberGym**: CyberGym is a public benchmark for measuring automated vulnerability discovery in real-world open-source software. It proxies the Cyber 2 track, focusing on a model's ability to identify and exploit vulnerabilities within complex, real-world codebases. Our evaluation tracks the mean vulnerability discovery rate averaged over multiple test cases to assess the model's precision in security analysis.
- **CyberBench**: CyberBench is an adversarial robustness benchmark designed to evaluate a model's ability to resist jailbreaking and safeguard bypass attempts from malicious actors. It measures the Attack Success Rate (ASR) across both single-turn and multi-turn adversarial scenarios, assessing the model's resilience in refusing harmful requests when exposed to sophisticated, iterative prompts. Our evaluation tracks both the refusal rate and the quality of safety-compliant responses, with results averaged over independent adversarial testing runs to ensure consistency.

General Capabilities and Reasoning

- **IFBench:** IFBench is an instruction-following benchmark that tests whether models can satisfy specific verifiable constraints embedded in prompts (e.g., word positioning, counting constraints, repetition patterns). The benchmark contains 294 tasks and we report mean pass rate across tasks. We average results across 4 runs to reduce variance. We report internal run results for Qwen3.6-27B and our model, and self-reported number for Gemma4-31B.
- **AIME 2026:** We use the full 30-question datasets for AIME 2026. This benchmark features highly challenging, non-multiple-choice mathematics problems from the American Invitational Mathematics Examination. The questions require creative problem-solving, advanced algebraic manipulation, and deep number theory skills, where the final answer is always an integer between 000 and 999. Results are averaged over 10 runs to reduce variance. We report internal run results for our model, and self-reported numbers for Gemma4-31B and Qwen3.6-27B.
- **GPQA Diamond:** We use the full 198-question dataset. This subset contains high-quality, multiple-choice questions in biology, physics, and chemistry that were validated by domain experts to ensure they cannot be easily looked up on Google. The problems require deep, graduate-level scientific reasoning and rigorous technical understanding to solve. Results are sourced from Artificial Analysis following the documented benchmarking methodology.
- **Humanity's Last Exam (HLE)**: This benchmark consists of highly advanced, crowd-sourced questions spanning dozens of subjects that are specifically designed to be difficult even for human experts to answer. The questions test advanced scientific knowledge, complex abstract reasoning, and niche academic concepts. No tool use is permitted. We use the text-only subset of 2,158 questions. Results are sourced from Artificial Analysis following the documented benchmarking methodology.
- **AA-LCR:** AA-LCR evaluates long-context performance by testing reasoning across multiple long documents. The questions are hard and text-based requiring genuine reasoning rather than simple data extraction. We use the full 100-question dataset spanning several document categories, such as academic papers, company financials, government consultations, legal documents, industry reports, and marketing materials. No tool use is permitted. Results are sourced from Artificial Analysis following the documented benchmarking methodology.
- **BEAM-128K:** BEAM is a text-only, non-agentic long-context memory evaluation comprising 400 conversations spanning 128K token length across various domains, such as coding and math. It is designed to assess ten distinct memory abilities (e.g., factual recall, temporal reasoning, state tracking). Each sample is evaluated as a direct, single-turn recall task: the entire conversation history is placed in-context and the model must answer a memory-probing question without any scaffold or tool use. Grading is LLM-judged using GPT-5.4, which scores model responses against reference answers. We measure mean pass@1 across four attempts. We report internal run results for all 3 models.