MMPostTrainBench: Benchmarking Autonomous Research for Multimodal Post-Training

Anonymous authors
Paper under double-blind review

Abstract

Autonomous research seeks sustained model improvements through iterative experimentation and feedback. LLM agents show promise in automating machine learning and language-model post-training, but their ability to sustain multimodal improvement remains unclear. We introduce MMPostTrainBench, a benchmark spanning eight tasks in image, audio, video, and joint audio-video understanding and image-grounded software repair. Agents operate from a common base model within fixed budgets, using development feedback before independent evaluation of their submitted models. Evaluation covers target and non-target model outcomes, iterative model improvement and selection, and research integrity. Across all eight tasks, 52.1% of model–task means fall below the base, and evaluated submissions also exhibit non-target regressions. Model performance does not consistently improve across research iterations, and agents do not reliably select the best evaluated candidate for submission; final submissions trail that candidate by up to 5.38 percentage points. Extending autonomous research from text-only to multimodal tasks introduces additional sources of error in perception, cross-modal alignment, and temporal grounding. The observed regressions and selection gaps highlight the need to balance targeted improvements with non-target capability preservation and to retain gains across research iterations. These requirements motivate MMResearch, a multimodal research framework that connects media-grounded evidence to hypotheses and interventions, carries findings across rounds through hierarchical memory, and retains candidates using development evaluation. Added to existing code-agent runtimes, it improves submitted-model accuracy by up to 7.75 percentage points for Claude Opus 4.8 with Claude Code and 2.33 points for GPT-5.6-sol with Codex.

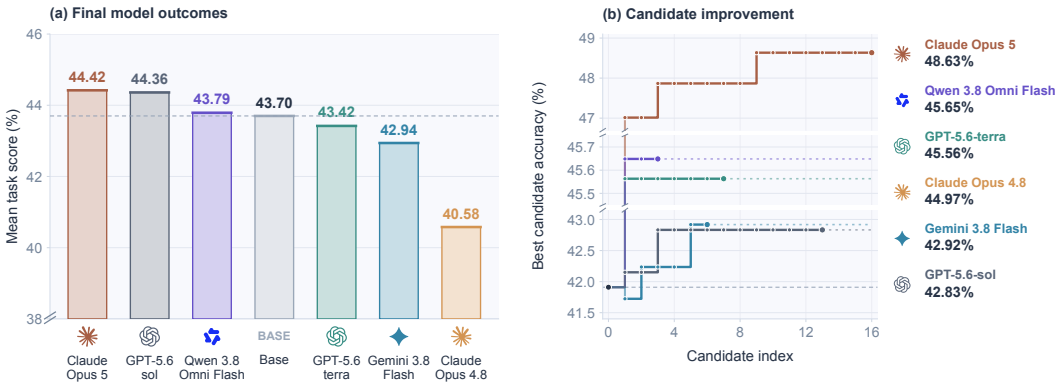


Figure 1: **MMPostTrainBench at a glance.** (a) **Final model outcomes:** post-training effectiveness measured by mean task score across all eight tasks relative to the common base model (Table 2). (b) **Iterative model improvement:** best-so-far held-out candidate accuracy on MMMU-Pro; labels report the best candidate score.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 Introduction

A central challenge in autonomous machine-learning research is to translate experimental feedback into effective model improvements and accumulate useful knowledge across successive experiments. Research agents tackle this challenge with code, tools, and feedback for experiments, language-model post-training, and training-data research (Huang et al., 2024; Chan et al., 2025; Wijk et al., 2024; Lu et al., 2024; Rank et al., 2026; Meng et al., 2026). Beyond text-only post-training, agents must connect evidence in source media with language and other modalities. Clutter, background sounds, and incomplete descriptions in media collected in the wild make relevant evidence difficult to isolate. This raises a broader question: *Can research agents autonomously improve how models see, hear, and reason about multimodal inputs—and sustain those improvements as experimentation continues?*

Autonomous multimodal post-training couples three challenges. **(1) Multimodal Error Diagnosis.** In visual reasoning, chain-of-thought prompting can degrade performance on perception-heavy tasks (Jiang et al., 2025a). Agents must inspect source media to distinguish missed or misinterpreted evidence from faulty inference before choosing an intervention. **(2) Data Curation and Modality Alignment.** Selecting visual instructions for vulnerable sample groups can improve robustness (Yang et al., 2025b). Agents must also preserve semantic and temporal alignment: labels should match observable content, and paired audio-video segments should retain their timing. **(3) Budgeted Multimodal Experimentation.** Autonomous language-model post-training evaluations report early termination without valid final checkpoints (Rank et al., 2026). Multimodal runs must also budget for frame sampling and audio processing alongside training, candidate evaluation, and final delivery. Effective research must therefore connect media evidence and alignment decisions to the improvements ultimately delivered.

We introduce MMPostTrainBench (Multimodal Post-Training Benchmark), which evaluates autonomous research on eight tasks spanning image, audio, video, and joint audio-visual understanding, as well as multimodal code tasks (Table 1; Figure 2). Task queries specify a target capability and benchmark, such as audio reasoning on MMAR, and require training pipelines to process the corresponding media. Under a common resource budget, agents select data sources, modality mixtures, supervision, and training strategies in response to development feedback. Harbor-compatible task packages provide this feedback during research and reserve separate held-out tests for final evaluation. Target and non-target evaluations assess whether a targeted update preserves other capabilities.

We assess **model outcomes**, **iterative model improvement**, and **research integrity**. Across six research-agent configurations on eight tasks, 52.1% of model–task means fall below the base, and submissions also exhibit non-target regressions. Figure 1 summarizes final outcomes and best-so-far candidate improvement. Final submissions trail the best evaluated candidates by up to 5.38 percentage points across recorded candidate pools. Integrity checks flag suspected misuse of evaluation data during training. Producing a strong candidate, selecting it reliably, and retaining other multimodal capabilities thus remain distinct requirements.

These findings motivate our proposed multimodal research framework, MMResearch. Building on persistent research orchestration (Jin et al., 2026), it augments existing code-agent runtimes with **multimodal evidence reasoning**, **hierarchical memory**, and **evaluation-guided model selection**. Observations are anchored to source media and linked to hypotheses, interventions, and development outcomes; memory carries successful recipes and failure records across rounds; model selection retains a candidate for final delivery. These mechanisms connect diagnosis, accumulated knowledge, and submission decisions within a seven-step research loop. In the evaluated configurations, adding MMResearch improves submitted-model accuracy by up to 7.75 percentage points for Claude Opus 4.8 with Claude Code and 2.33 points for GPT-5.6-sol with Codex.

Our contributions are threefold: **(1)** We introduce MMPostTrainBench, which evaluates autonomous post-training across eight multimodal tasks through model outcomes, iterative model improvement, and research integrity. **(2)** We characterize non-target regressions, candidate degradation, and submission-selection failures in current research agents. **(3)** We propose MMResearch, which combines multimodal evidence reasoning, hierarchical memory, and evaluation-guided model selection to improve submitted models within existing code-agent runtimes.

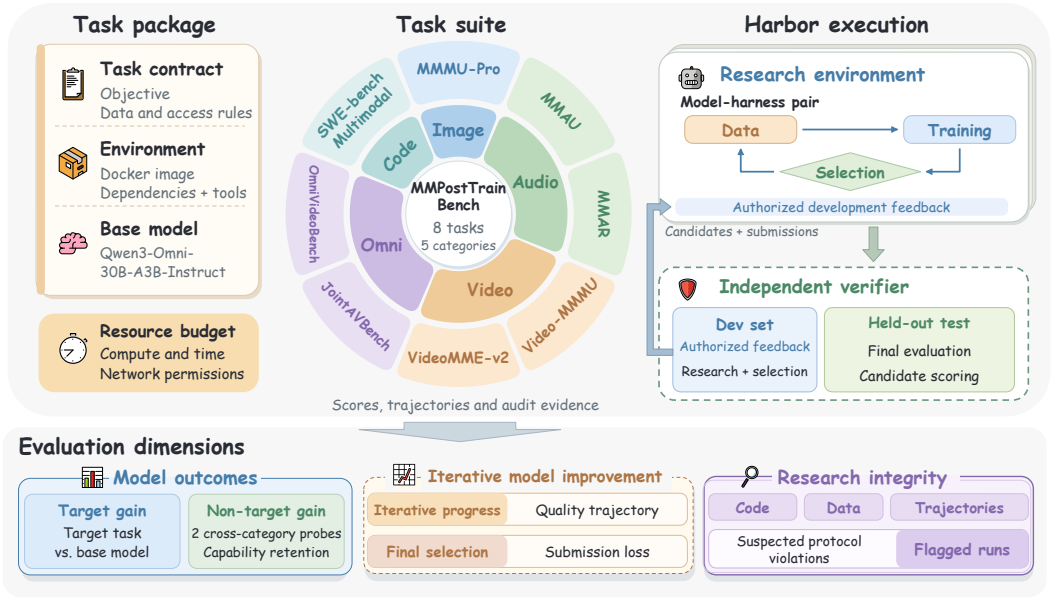


Figure 2: **MMPostTrainBench benchmark overview.** Agents conduct post-training from a common base model on eight multimodal tasks within fixed budgets. To limit evaluation leakage, execution and verification run in separate containers, research feedback is restricted to authorized development data, and final tests remain sealed. Submitted models, candidate trajectories, and audits support evaluation of model outcomes, iterative model improvement, and research integrity.

2 Related Work

Autonomous Research Benchmarks. Autonomous research benchmarks evaluate progress at different levels of an AI system. Task-level evaluations measure agents' ability to conduct ML experiments and solve AI R&D problems (Huang et al., 2024; Chan et al., 2025; Wijk et al., 2024). Other evaluations focus on the supporting harness and infrastructure (Wu et al., 2026c; Ding et al., 2026), or on the agent's ability to learn from broad goals and interaction feedback (Wu et al., 2026b; Shi et al., 2026). For autonomous post-training, the object of improvement is the target model: PostTrainBench permits end-to-end language-model post-training, whereas RSIBench-Data fixes the learning stack to isolate training-data research (Rank et al., 2026; Meng et al., 2026). MMPostTrainBench extends this setting to multimodal tasks, assessing final target and non-target performance, iterative candidate quality and submission choice, and research integrity.

Research Agent Harnesses. Research agent harnesses must organize experiments and make accumulated evidence useful for subsequent decisions. End-to-end workflows coordinate idea generation, implementation, and training (Lu et al., 2024; Zhang et al., 2026a), while search-based systems structure the choice of experiments (Jiang et al., 2025b). As experimentation continues, preserving and reusing research state becomes equally important. Hierarchical knowledge retrieval and persistent hypothesis trees connect earlier findings to later decisions (Zhang et al., 2026b; Jin et al., 2026); discovery replay and reusable environment memory serve a related role in autonomous exploration (Zheng et al., 2026; Zhu et al., 2026). ModularRSI refactors Terminus-2 into behavioral modules and evolves their source functions (Wu et al., 2026a). Building on research orchestration and knowledge reuse, MMResearch focuses on improving multimodal models through post-training, connecting multimodal evidence-based diagnosis with training interventions and model selection.

3 Method

MMPostTrainBench evaluates autonomous multimodal post-training through a controlled research environment and three complementary evaluation dimensions (Figure 2). We describe the task setting and evaluation protocol before introducing our proposed research framework, MMResearch.

3.1 Task Definition and Research Process

Research task. A task assigns the agent a capability to improve, a starting model, and a bounded research environment. Following the artifact-oriented setup of autonomous post-training and research benchmarks (Rank et al., 2026; Meng et al., 2026; Jin et al., 2026), we represent task b as

$$\mathcal{T}_b = (q_b, \theta_0, \mathcal{D}_b^{\text{adm}}, F_b^{\text{dev}}, E_b^{\text{test}}, \mathcal{B}_b). \tag{1}$$

Here q_b specifies the target capability and success criterion; θ_0 is the common base model; $\mathcal{D}_b^{\text{adm}}$ specifies permitted training-data sources and uses; F_b^{dev} returns authorized development feedback; E_b^{test} independently evaluates frozen models; and $\mathcal{B}_b$ specifies compute allocation and the research time limit. Research configuration a consists of a frontier model and its execution runtime.

Autonomous research. Within loop r, the agent chooses interventions from its accumulated history rather than following a prescribed training recipe. Suppressing a, b, r for clarity, candidate production and feedback can be written as

$$\begin{aligned} u_t &= \pi_a(q_b, \mathcal{H}_{<t}), & \theta_t &= \text{Execute}(u_t; \theta_0, \mathcal{C}_{<t}), \\ f_t &= F_b^{\text{dev}}(\theta_t), & \mathcal{H}_{\le t} &= \mathcal{H}_{<t} \cup \{(u_t, \theta_t, f_t, \ell_t)\}. \end{aligned} \tag{2}$$

An intervention u_t specifies data construction, a training recipe, or a model-side operation using permitted prior candidates $\mathcal{C}_{<t}$; ℓ_t records execution diagnostics and provenance. The initial history contains permitted task materials, and later observations include only information available during research. The agent decides how many experiments to attempt and which completed model $S_{a,b,r} = \theta_{a,b,r}^{\text{sub}}$ to submit before the budget expires.

Table 1: Eight post-training tasks, their domains, and target model abilities.

Task	Domain	Model Ability
MMMU-Pro (Yue et al., 2025)	Image	Image-grounded multidisciplinary reasoning
MMAU (Sakshi et al., 2025)	Audio	Audio understanding and reasoning
MMAR (Ma et al., 2025)	Audio	Audio reasoning
Video-MMMU (Hu et al., 2025)	Video	Video-grounded multidisciplinary reasoning
VideoMME-v2 (Fu et al., 2026)	Video	Video understanding
JointAVBench (Chao et al., 2026)	Omni	Joint audio-video understanding
OmniVideoBench (Li et al., 2026)	Omni	Joint audio-video understanding and reasoning
SWE-bench Multimodal (Yang et al., 2025a)	Code	Image-grounded software repair

3.2 Research Environment and Protocol

Task environment. Each Harbor-compatible task package specifies the task objective, designated base model, data-use rules, tool interfaces, and resource limits. The research environment provides a writable workspace and the dependencies required for model training and evaluation. Within the task's permissions, agents can inspect multimodal media, construct training examples, modify training code and configurations, execute experiments, and retain candidate models. Harbor's separate-verifier interface and the Docker driver separate research execution from independent verification.

Development feedback. Agents query F_b^{dev} to obtain candidate scores, permitted evaluation examples, and execution diagnostics. These observations support error diagnosis and subsequent decisions about data construction, training configurations, and candidate selection. The task's data-use rules determine which materials may serve as training supervision; access to evaluation examples for diagnosis does not itself authorize their use for training. Development feedback excludes final held-out test instances, reference answers, and scores, which remain withheld during research.

Submission and verification. At the end of research, the selected model's weights, configuration, and loading dependencies are frozen and transferred to the independent verifier. The final evaluator

E_b^{test} assesses the base and submitted models using the same fixed held-out test instances, preprocessing, inference interface, and scoring procedure. This matched evaluation provides the basis for measuring changes in model performance. Data provenance, tool traces, and candidate records connect the submitted model to its research process and support the integrity checks defined in Section 3.3.

3.3 Evaluation Dimensions

We evaluate three complementary dimensions: model outcomes, iterative model improvement, and research integrity. Let $S = \theta_{a,b}^{\text{sub}}$ denote the submitted model and $H_b(\theta) \in [0, 1]$ its raw task score: answer accuracy for understanding tasks and resolved-instance rate for SWE-bench Multimodal.

Model outcomes. We compare submissions with the base on the target task and two prespecified non-target probes from other categories. Target change is $\Delta_{a,b} = 100[\widehat{s}_{a,b} - H_b(\theta_0)]$ percentage points, where $\widehat{s}_{a,b}$ is the mean of three loop scores, with flagged loops reset to the base before averaging. Non-target changes assess transfer and capability retention. Aggregate comparisons weight all eight tasks equally. For SWE-bench Multimodal, Table 2 reports submitted-model means relative to a base resolved-instance rate of 2.29%.

Iterative model improvement. We assess iterative model improvement from two perspectives. **(1) Iterative progress:** whether further experiments improve candidate quality or lead to stagnation, degradation, or fluctuations, as reflected by candidate trajectories evaluated retrospectively on the same held-out test set. **(2) Final selection:** whether the final submission retains the best performance among the evaluated candidates produced before submission. Let $\mathcal{C}_{a,b}^{\text{rec}}$ denote this candidate set. Submission loss is defined as

$$R_{a,b}^{\text{rec}} = 100 \left[\max_{\theta \in \mathcal{C}_{a,b}^{\text{rec}} \cup \{S\}} H_b(\theta) - H_b(S) \right]. \tag{3}$$

A positive value indicates that the final submission underperforms the best evaluated candidate, with the gap measured in percentage points.

Research integrity. Benchmark scores may be inflated by contamination or misuse of evaluation information. Audit categories cover training-data contamination, unauthorized workspace or data access, prohibited split use, and misuse of evaluation information. Retained code, data provenance, and research trajectories provide evidence for these checks. Results are summarized by the number of runs flagged for suspected violations.

3.4 MMResearch: Multimodal Research Framework

MMResearch adds a research controller to existing code-agent runtimes, retaining their native tools and action loops while coordinating experiments, memory, and model delivery. Figure 3 shows how hierarchical memory (a) and multimodal evidence (c) support the seven-step research loop (b), which concludes with model delivery (d). The evidence-anchor and graph design in (c) specifies how localized media observations connect to research hypotheses and experimental outcomes.

Seven-Step Research Loop. Each round turns a research direction into an evaluated candidate and a decision (Figure 3b). Select chooses a data- or model-side direction from prior hypotheses and failure records. Diagnose inspects authorized multimodal inputs to investigate the suspected failure. Debate reviews the diagnosis, intervention, feasibility, and risks. Implement executes the intervention and saves a candidate. Evaluate screens and confirms candidates on development data. Record stores configurations, results, and rationales in hierarchical memory. Decide uses confirmation results to promote the candidate, refine the approach, or pivot to another direction; a proxy improvement alone does not justify promotion. A separate checkpoint tracks the retained model, initially the base. Before the budget expires, the controller checks required records and freezes the retained model's weights and configuration for delivery.

Multimodal Evidence. The agent inspects authorized image, audio, video, and audio-video inputs through native perception or configured tools. In the *multimodal evidence graph* design (Figure 3c), each *evidence anchor* identifies a source sample, its modality, and a relevant region or time span. When finer localization is unavailable, the anchor refers to the whole image or clip. Direct observations are stored separately from the hypotheses inferred from them. Links connect observations, hypotheses, training interventions, and development-evaluation outcomes, making diagnoses traceable

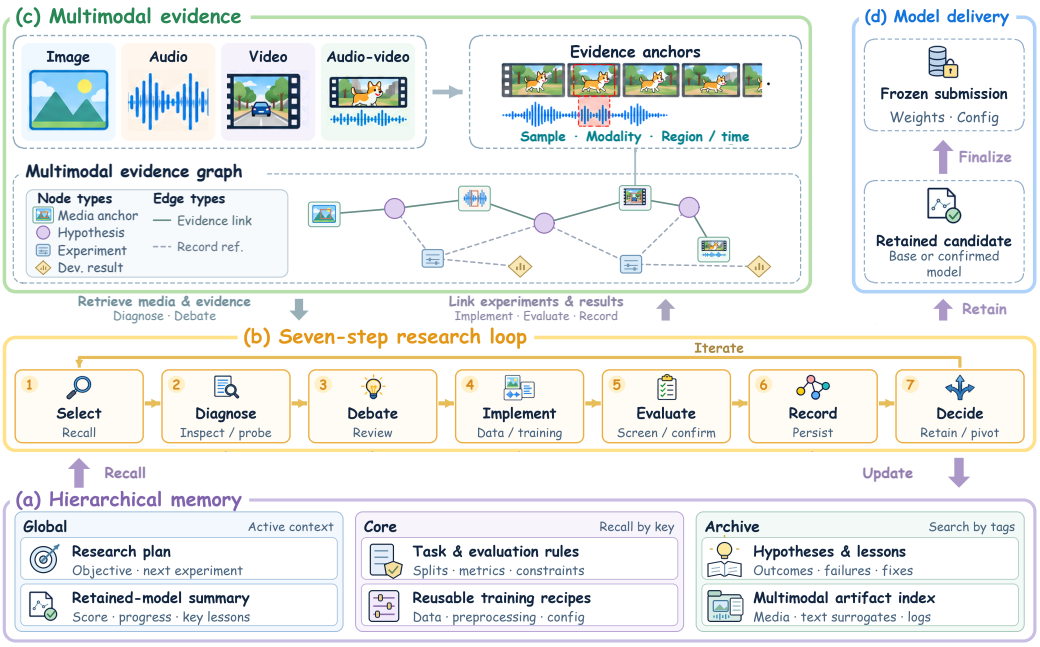


Figure 3: **MMResearch framework.** MMResearch coordinates a seven-step research loop (b), using hierarchical memory (a) to guide experimentation and multimodal evidence (c) to connect observations with experimental decisions, then freezes the retained candidate for delivery (d).

to the underlying media. DIAGNOSE and DEBATE use the linked observations to review explanations; RECORD preserves the evidence and outcome links for subsequent rounds. Later rounds can follow these links back to the source media and prior interventions to reassess an earlier diagnosis in light of development results. Held-out test results remain outside this research feedback loop.

Hierarchical Memory. MMResearch preserves research context across rounds in three memory tiers (Figure 3a). *Global* maintains the active research plan and a summary of the retained model. *Core* stores task and evaluation rules and reusable training recipes. *Archive* indexes hypotheses, outcomes, failures, and media artifacts. SELECT retrieves relevant records at the start of a round, and RECORD updates them with its results and lessons, allowing later experiments to build on earlier findings.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Tasks and Resources. All loops start from Qwen3-Omni-30B-A3B-Instruct (Xu et al., 2025) on the eight-task suite spanning the Image, Audio, Video, Omni, and Code domains (Table 1). Training uses eight GPUs, with a default limit of 24 hours per loop.

Research Agents. We evaluate six research-model/runtime configurations: GPT-5.6-sol and GPT-5.6-terra with Codex; Claude Opus 5 and Claude Opus 4.8 with Claude Code; Gemini 3.8 Flash with Gemini CLI; and Qwen 3.8 Omni Flash with Qwen Code. These models direct experiments through their native runtimes, while Qwen3-Omni is the shared model being post-trained.

Evaluation Protocol. For each model–benchmark pair, we run three independent loops with fresh histories, the same base model, and the same budget. Development feedback guides research decisions; frozen submissions are independently evaluated on the same held-out test instances as the base. Each loop is audited; a loop triggering a reset contributes the base score, otherwise its submitted-model score. Table 2 reports the three-loop mean. We select one of the three loops per pair for non-target evaluation and candidate analysis.

4.2 Main Results

Model outcomes. Across all eight tasks, 52.1% of the 48 model–task means fall below the base; the largest mean gain is 0.73 pp (Table 2). Qwen 3.8 Omni Flash gains 2.80 pp on MMMU-Pro but loses 2.26 pp on OmniVideoBench. Gemini 3.8 Flash leads on MMAR yet regresses on both video tasks. Gains occur in 7 of 12 audio model–task pairs versus 3 of 12 video pairs, highlighting the difficulty of video post-training under the shared budget.

Table 2: **Final model outcomes on MMPostTrainBench.** Three-loop mean task score (%) and change from base (pp). Harnesses appear below model names; bold marks the best result per row. The aggregate weights all eight tasks equally.

Category	Benchmark	Base model	Claude Opus 5 Claude Code	GPT-5.6 sol Codex	Qwen 3.8 Omni Flash Qwen Code	GPT-5.6 terra Codex	Gemini 3.8 Flash Gemini CLI	Claude Opus 4.8 Claude Code
Image	MMMU-Pro	41.91	43.26 +1.35	41.88 -0.03	**44.71** **+2.80**	43.00 +1.09	41.88 -0.03	41.72 -0.19
Audio	MMAU	48.60	50.25 +1.65	**50.30** **+1.70**	50.00 +1.40	49.40 +0.80	50.00 +1.40	44.01 -4.59
	MMAR	72.40	72.16 -0.24	72.31 -0.09	72.01 -0.39	73.00 +0.60	**73.50** **+1.10**	66.77 -5.63
Video	Video-MMMU	60.00	**60.89** **+0.89**	60.40 +0.40	53.96 -6.04	58.42 -1.58	49.01 -10.99	56.44 -3.56
	VideoMME-v2	24.81	23.40 -1.41	24.35 -0.46	**25.81** **+1.00**	22.49 -2.32	23.21 -1.60	22.89 -1.92
Omni	JointAVBench	59.94	59.91 -0.03	59.96 +0.02	63.90 +3.96	59.97 +0.03	**64.91** **+4.97**	58.84 -1.10
	OmniVideoBench	39.61	42.40 +2.79	**42.99** **+3.38**	37.35 -2.26	39.02 -0.59	38.57 -1.04	31.86 -7.75
Code	SWE-bench Multimodal	2.29	**3.13** **+0.83**	2.71 +0.42	2.57 +0.28	2.08 -0.21	2.43 +0.14	2.08 -0.21
Mean task score (%)		43.70	**44.42** **+0.73**	44.36 +0.67	43.79 +0.09	43.42 -0.27	42.94 -0.76	40.58 -3.12

Cross-modal capability retention. Non-target evaluation reveals capability loss beyond the optimization target (Table 3). For example, Qwen's MMMU-Pro submission loses 9.88 pp on MMAR. All six SWE-bench Multimodal configurations improve MMMU-Pro scores by 0.12–1.67 pp while reducing MMAR scores by 0.20–1.00 pp (Appendix B.1). These results show that image-grounded software-repair research can also affect audio understanding. Target accuracy and non-target retention therefore capture distinct aspects of post-training quality.

Table 3: **Cross-modal capability retention.** Cells show mean accuracy changes over the two indicated non-target probes (pp): Image/MMMU-Pro, Audio/MMAR, and Omni/JointAVBench. Underlining indicates the best result in each row.

Target benchmark	Non-target modalities	Claude Opus 5	GPT-5.6 sol	GPT-5.6 terra	Gemini 3.8 Flash	Qwen 3.8 Omni Flash	Claude Opus 4.8
MMMU-Pro	Audio, Omni	+4.08	+0.84	-0.51	+2.78	-4.40	-8.45
MMAR	Image, Omni	-0.98	-0.79	-0.57	-0.15	+0.43	+1.35
MMAU	Image, Omni	+1.71	-1.06	+1.26	-1.43	+0.04	+4.75
Video-MMMU	Image, Audio	+1.18	+1.01	+0.39	+1.86	-1.17	-2.95
VideoMME-v2	Image, Audio	+0.82	+0.76	-0.42	-1.08	-1.50	-0.08
JointAVBench	Image, Audio	-0.85	+0.90	+0.34	-15.31	-3.37	-5.24
OmniVideoBench	Image, Audio	-0.28	+0.50	+0.84	+0.00	-0.08	-11.68
SWE-bench Multimodal	Image, Audio	+0.34	-0.29	-0.24	+0.24	+0.50	-0.31
Overall		+0.75	+0.23	+0.14	-1.64	-1.20	-2.83

Resource Usage. Figure 4 reports mean loop duration and agent token usage for the existing run set. GPT-5.6-terra averages 23.7 hours under the 24-hour budget, yet its mean task score remains 0.27 pp below the base (Table 2). Claude Opus 5 and GPT-5.6-sol achieve similar eight-task mean scores (44.42% and 44.36%), while Opus 5 uses about 31% fewer tokens. Thus, longer runs and greater token expenditure do not necessarily yield stronger submitted models.

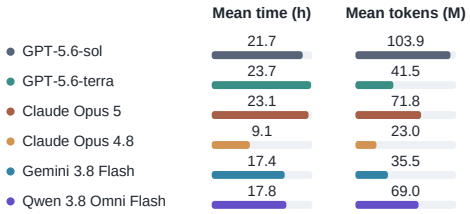


Figure 4: Mean time and tokens.

Iterative model improvement. Further experimentation can degrade candidates, and final submissions trail the best evaluated candidate by up to 5.38 pp (Figure 5; Table 4). In the selected Qwen 3.8 Omni Flash loop on MMMU-Pro, candidates score 45.65%, 44.71%, and 43.94% on the held-out test. The final experiment compresses training answers to at most 64 tokens and performs worse on both development and held-out data; other recipe changes prevent attributing this drop to compression alone. Development results are 62/137, 62/137, and 59/137, respectively. Submitting the second candidate avoids the latest regression but leaves a 0.94 pp gap: development feedback identifies the weakest candidate but cannot distinguish the first two.

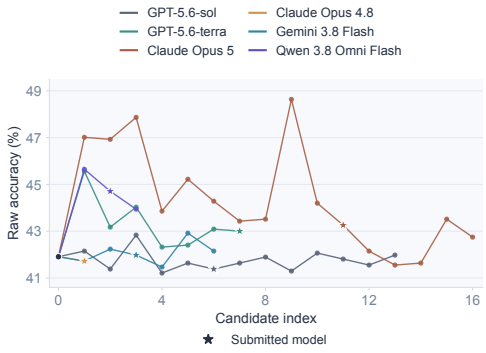


Figure 5: **MMMU-Pro candidate accuracy.**

Table 4: **Submission loss.** R^{rec} is best-prior minus submitted accuracy (pp).

Research model	Best prior (%)	Submitted (%)	$R^{\text{rec}}\downarrow$ (pp)
Gemini 3.8 Flash	42.92	41.98	0.94
Qwen 3.8 Omni Flash	45.65	44.71	0.94
GPT-5.6-sol	42.83	41.38	1.45
GPT-5.6-terra	45.56	43.00	2.56
Claude Opus 4.8	44.97	41.72	3.25
Claude Opus 5	48.63	43.26	5.38

Research integrity. Observed violations include evaluation-data contamination and exploitation of answer-label patterns. Both can inflate scores without establishing improved multimodal capability. Figure 6 records ten flagged runs across six models and four categories. Four involve JointAVBench: GPT-5.6-terra, Claude Opus 4.8, Gemini, and Qwen. Of its 1,948 reference answers, 91.58% are D. Gemini and Qwen predict D in 65.91% and 72.64% of cases, with macro recall of 45.15% and 48.90%, respectively. Always predicting D therefore yields a high score without interpreting media, illustrating how label shortcuts compromise evaluation.

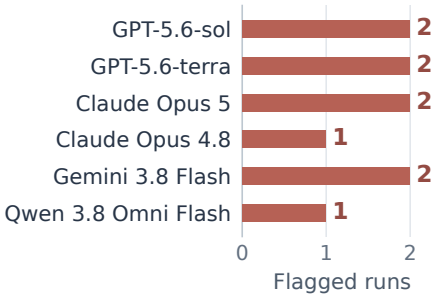


Figure 6: **Recorded integrity flags.**

4.3 MMResearch Evaluation

Table 5 evaluates MMResearch with Claude Opus 4.8/Claude Code and GPT-5.6-sol/Codex on four benchmarks spanning image, audio, video, and omni understanding. Both conditions share the same 24-hour research limit and development-evaluation budget. **MMResearch improves submission accuracy in seven of eight comparisons**: all four for Opus 4.8 and three for GPT-5.6-sol. Their MMMU-Pro gains are 2.02 and 2.33 pp, respectively.

Table 5: **Effect of MMResearch on submission quality.** Accuracy (%) measures final model performance; parentheses report gains over the same harness without MMResearch (pp).

Research model	Harness	MMMU-Pro *Image*	MMAR *Audio*	VideoMME-v2 *Video*	OmniVideoBench *Omni*
Claude Opus 4.8	Claude Code	41.72	66.77	22.89	31.86
	+ MMResearch	43.74 (+2.02)	72.40 (+5.63)	24.81 (+1.92)	39.61 (+7.75)
GPT-5.6-sol	Codex	41.88	72.31	24.35	42.99
	+ MMResearch	44.21 (+2.33)	72.90 (+0.60)	24.81 (+0.46)	42.61 (-0.38)
Base model (main-table reference)		41.91	72.40	24.81	39.61

With MMResearch, all eight scores match or exceed the base. Opus 4.8 matches it on MMAR, VideoMME-v2, and OmniVideoBench; both research models exceed it on MMMU-Pro. These results show that MMResearch improves submission quality through both recovery from degradation and gains beyond the base.

4.4 Ablation Studies

Development Feedback Budget. To examine how development feedback affects autonomous research, we evaluate Claude Opus 4.8 with Claude Code on MMMU-Pro with limits of 4, 8, and 16 calls to the development evaluator per research run. Each call returns feedback to guide subsequent research. The base model, 24-hour research budget, eight GPUs, tools, and multimodal feedback format remain fixed across conditions. Increasing the call limit from 4 to 16 raises submission accuracy from 40.56% to 42.09%, with smaller incremental gains at the higher budget (Figure 7).

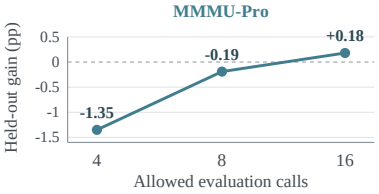


Figure 7: **Development feedback budget on MMMU-Pro.**

MMResearch Components. Using the same model–harness pair and benchmark, we assess the contributions of hierarchical memory and multimodal evidence within MMResearch. *w/o Hierarchical Memory* removes structured storage and retrieval across rounds; *w/o Multimodal Evidence* removes evidence organization and media-grounded diagnosis. Native context, raw media access, and candidate-selection rules remain unchanged. Removing hierarchical memory or multimodal evidence lowers accuracy from 43.74% to 42.10% and 42.83%, respectively (Table 6). The drops of 1.64 and 0.91 pp support both components' contribution to final submission quality in this setting.

Table 6: **MMResearch component ablations on MMMU-Pro.** Gain is the accuracy change from the base (pp).

Variant	Gain ↑
Full	+1.83
w/o Hierarchical Memory	+0.19
w/o Multimodal Evidence	+0.92

5 Conclusion

We introduced MMPostTrainBench, a benchmark for autonomous post-training across image, audio, video, and joint audio-video understanding, as well as image-grounded software repair. It evaluates model outcomes, iterative model improvement, and research integrity. The results reveal unreliable target gains, non-target capability regressions, candidate degradation and selection failures, and suspected protocol violations. These findings motivate MMResearch, which combines multimodal evidence reasoning, hierarchical memory, and evaluation-guided model selection to support more reliable research. Compared with unaugmented harnesses, MMResearch improves submission accuracy in seven of eight model–task comparisons across four modalities. Future work should investigate how agents can use multimodal evidence to improve target capabilities while preserving other modalities, and develop research strategies that transfer across tasks.

AI USE STATEMENT

We used generative AI to assist with result analysis, manuscript drafting, and language polishing. Separately, the research agents evaluated in MMPostTrainBench used language models to conduct autonomous post-training research, including data construction, training, and candidate selection, as described in the experimental protocol. Benchmark performance was measured using task-specific evaluation procedures. The authors retain full responsibility for the methodology, reported results, interpretations, and final manuscript.

REPRODUCIBILITY STATEMENT

The paper describes the MMResearch framework, benchmark tasks, environment configurations, resource budgets, and evaluation protocols. We retain framework configurations, research records, and scripts for figure generation and result verification to support reproducibility.

REFERENCES

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Aleksander Madry, and Lilian Weng. MLE-bench: Evaluating machine learning agents on machine learning engineering. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=6s5uXNWGIh.

Jianghan Chao, Jianzhang Gao, Wenhui Tan, Yuchong Sun, Ruihua Song, and Liyun Ru. JointAVBench: A benchmark for joint audio-visual reasoning evaluation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Zg1YH8R5GG.

Leilei Ding, Shumin Wang, Yuting Huang, Fanqi Wan, Yinmin Zhang, Qi Han, Yiming Xu, Feiyuan Zhang, Xiaomeng Chu, Guoliang You, Wuyang Zhang, Daxin Jiang, and Yanyong Zhang. Φ-Bench: Can large language models engineer the infrastructure that powers them? arXiv:2609.10226, 2026. URL https://arxiv.org/abs/2609.10226.

Chaoyou Fu, Haozhi Yuan, Yuhao Dong, Yi-Fan Zhang, Yunhang Shen, Xiaoxing Hu, Xueying Li, Jinsen Su, Chengwu Long, Xiaoyao Xie, Yongkang Xie, Xiawu Zheng, Xue Yang, Haoyu Cao, Yunsheng Wu, Ziwei Liu, Xing Sun, Caifeng Shan, and Ran He. Video-MME-v2: Towards the next stage in benchmarks for comprehensive video understanding. arXiv:2604.05015, 2026. URL https://arxiv.org/abs/2604.05015.

Kairui Hu, Penghao Wu, Fanyi Pu, Wang Xiao, Yuanhan Zhang, Xiang Yue, Bo Li, and Ziwei Liu. Video-MMMU: Evaluating knowledge acquisition from multi-discipline professional videos. arXiv:2501.13826, 2025. URL https://arxiv.org/abs/2501.13826.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. MLAgentBench: Evaluating Language Agents on Machine Learning Experimentation. In International Conference on Machine Learning (ICML), 2024. URL https://arxiv.org/abs/2310.03302.

Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanwei Li, Yu Qi, Xinyan Chen, Liuhui Wang, Jianhan Jin, Claire Guo, Shen Yan, Bo Zhang, Chaoyou Fu, Peng Gao, and Hongsheng Li. MME-CoT: Benchmarking chain-of-thought in large multimodal models for reasoning quality, robustness, and efficiency. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 27793–27830. PMLR, 2025a. URL https://proceedings.mlr.press/v267/jiang25n.html.

Zhengyao Jiang, Dominik Schmidt, Dhruv Srikanth, Dixing Xu, Ian Kaplan, Deniss Jacenko, and Yuxiang Wu. AIDE: AI-Driven Exploration in the Space of Code. arXiv:2502.13138, 2025b. URL https://arxiv.org/abs/2502.13138.

Jiajie Jin, Yuyang Hu, Kai Qiu, Qi Dai, Chong Luo, Guanting Dong, Xiaoxi Li, Tong Zhao, Xiaolong Ma, Gongrui Zhang, Zhirong Wu, Bei Liu, Zhengyuan Yang, Linjie Li, Lijuan Wang, Hongjin Qian, Yutao Zhu, and Zhicheng Dou. Toward generalist autonomous research via hypothesis-tree refinement. arXiv:2606.11926v1, 2026. URL https://arxiv.org/abs/2606.11926.

Caorui Li, Yu Chen, Yiyan Ji, Jin Xu, Zhenyu Cui, Shihao Li, Yuanxing Zhang, Zhenghao Song, Dingling Zhang, Ying He, Haoxiang Liu, Yuxuan Wang, Qiufeng Wang, Jiafu Tang, Zhenhe Wu, Jiehui Luo, Zhiyu Pan, Weihao Xie, Chenchen Zhang, Zhaohui Wang, Jiayi Tian, Yanghai Wang, Zhe Cao, Minxin Dai, Ke Wang, Runzhe Wen, Yinghao Ma, Yaning Pan, Sungkyun Chang, Termeh Taheri, Haiwen Xia, Christos Plachouras, Emmanouil Benetos, Yizhi Li, Ge Zhang, Jian Yang, Tianhao Peng, Zili Wang, Minghao Liu, Junran Peng, Zhaoxiang Zhang, and Jiaheng Liu. OmniVideoBench: Towards audio-visual understanding evaluation for omni MLLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ItRYEe8E61.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery. arXiv:2408.06292, 2024. URL https://arxiv.org/abs/2408.06292.

Ziyang Ma, Yinghao Ma, Yanqiao Zhu, Chen Yang, Yi-Wen Chao, Ruiyang Xu, Wenxi Chen, Yuanzhe Chen, Zhuo Chen, Jian Cong, Kai Li, Keliang Li, Siyou Li, Xinfeng Li, Xiquan Li, Zheng Lian, Yuzhe Liang, Minghao Liu, Zhikang Niu, Tianrui Wang, Yuping Wang, Yuxuan Wang, Yihao Wu, Guanrou Yang, Jianwei Yu, Ruibin Yuan, Zhisheng Zheng, Ziya Zhou, Haina Zhu, Wei Xue, Emmanouil Benetos, Kai Yu, Eng-Siong Chng, and Xie Chen. MMAR: A challenging benchmark for deep reasoning in speech, audio, music, and their mix. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=fgmrBJemlQ.

Fanqing Meng, Lingxiao Du, Qiguang Chen, Ziqi Zhao, Haocheng Lu, Mengkang Hu, and Michael Qizhe Shieh. RSIBench-Data: Benchmarking data-centric research for recursive self-improvement. arXiv:2607.25886, 2026. URL https://arxiv.org/abs/2607.25886.

Ben Rank, Hardik Bhatnagar, Ameya Prabhu, Shira Eisenberg, Karina Nguyen, Matthias Bethge, and Maksym Andriushchenko. PostTrainBench: Can LLM agents automate LLM post-training? arXiv:2603.08640v2, 2026. URL https://arxiv.org/abs/2603.08640.

S Sakshi, Utkarsh Tyagi, Sonal Kumar, Ashish Seth, Ramaneswaran Selvakumar, Oriol Nieto, Ramani Duraiswami, Sreyan Ghosh, and Dinesh Manocha. MMAU: A massive multi-task audio understanding and reasoning benchmark. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=TeVAZXr3yv.

Jiajun Shi, Siyuan Tao, Yuhao Wu, Zexuan Wang, Jingyuan Zhang, et al. S3Gym: Can LLMs turn self-testing and self-judging into self-improvement? arXiv:2608.31100, 2026. URL https://arxiv.org/abs/2608.31100.

Hjalmar Wijk, Tao Lin, Joel Becker, Sami Jawhar, Neev Parikh, Thomas Broadley, Lawrence Chan, Michael Chen, Josh Clymer, Jai Dhyani, Elena Ericheva, Katharyn Garcia, Brian Goodrich, Nikola Jurkovic, Holden Karnofsky, Megan Kinniment, Aron Lajko, Seraphina Nix, Lucas Sato, William Saunders, Maksym Taran, Ben West, and Elizabeth Barnes. RE-Bench: Evaluating frontier AI R&D capabilities of language model agents against human experts. arXiv:2411.15114, 2024. URL https://arxiv.org/abs/2411.15114.

Siwei Wu, Jincheng Ren, Yizhi Li, Haau-Sing Li, Chengran Yang, Yuxuan Zhang, Weicheng Gu, Jian Yang, Riza Batista-Navarro, Chuanyi Zhang, Xianglong Liu, Ming Zhou, Bryan Dai, and Chenghua Lin. ModularRSI: Modular and generalizable recursive harness self-improvement, 2026a. URL https://arxiv.org/abs/2609.14857v1.

Yuhao Wu, Jingyuan Zhang, Jiajun Shi, et al. Aspire: Can models self-evolve from vague goals? arXiv:2608.31111, 2026b. URL https://arxiv.org/abs/2608.31111.

Yuhao Wu, Jingyuan Zhang, Jiajun Shi, et al. HarnessDev: Can LLMs create and evolve their own agent harness? arXiv:2609.01437, 2026c. URL https://arxiv.org/abs/2609.01437.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, Baosong Yang, Bin Zhang, Ziyang Ma, Xipin Wei, Shuai Bai, Keqin Chen, Xuejing Liu, Peng Wang, Mingkun Yang, Dayiheng Liu, Xingzhang Ren,

Bo Zheng, Rui Men, Fan Zhou, Bowen Yu, Jianxin Yang, Le Yu, Jingren Zhou, and Junyang Lin. Qwen3-Omni Technical Report. arXiv:2509.17765, 2025. URL https://arxiv.org/abs/2509.17765.

John Yang, Carlos E Jimenez, Alex L Zhang, Kilian Lieret, Joyce Yang, Xindi Wu, Ori Press, Niklas Muennighoff, Gabriel Synnaeve, Karthik R Narasimhan, Diyi Yang, Sida Wang, and Ofir Press. SWE-bench multimodal: Do AI systems generalize to visual software domains? In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=riTiq3i21b.

Xu Yang, Chen Liu, and Ying Wei. Data selection matters: Towards robust instruction tuning of large multimodal models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 9326–9364, 2025b. URL https://openreview.net/forum?id=oLJMsGMfqr.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. MMMU-pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15134–15186. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.736. URL https://aclanthology.org/2025.acl-long.736/.

Ruiyi Zhang, Peijia Qin, Qi Cao, Li Zhang, and Pengtao Xie. AIBuildAI: An AI agent for automatically building AI models. *arXiv preprint arXiv:2604.14455*, 2026a. URL https://arxiv.org/abs/2604.14455.

Ruiyi Zhang, Peijia Qin, Qi Cao, Li Zhang, and Pengtao Xie. AIBuildAI-2: A knowledge-enhanced agent for automatically building AI models. *arXiv preprint arXiv:2605.27873*, 2026b. URL https://arxiv.org/abs/2605.27873.

Tong Zheng, Xidong Wu, Zheng Zhang, Zhankui He, Chaoyi Zhang, Benjamin Coleman, Ruoqiao Wei, Di Bai, Haolin Liu, Rui Liu, Xue Wang, Yue Zhuan, Wang-Cheng Kang, Renkai Xiang, Heng Huang, Xinwu Cheng, and Yunsong Guo. Dream-RSI: Recursive self-improvement through evolving worlds. arXiv:2609.14858v1, 2026. URL https://arxiv.org/abs/2609.14858.

Sibo Zhu, Shicheng Fan, Xinyue Wang, Wenyi Wu, Kun Zhou, and Biwei Huang. RSIAgent: Autonomous exploration for recursive self-improvement in new environments. arXiv:2609.15364v1, 2026. URL https://arxiv.org/abs/2609.15364.

A Task and Evaluation Details

Task suite and research setting. Table 7 lists the eight tasks, their development and test set sizes, and scoring measures. The understanding tasks use Qwen3-Omni-30B-A3B-Instruct as the post-training base. Training uses eight GPUs, with a default research limit of 24 hours per loop. The agent selects its data-construction procedure, training recipe, and sequence of experiments within the task's permitted resources. The reported model–benchmark target means use three independent research loops with fresh histories, the same base model, and the same per-loop budget.

Table 7: Task categories, development and test set sizes, and evaluation measures.

Task	Domain	Dev.	Test	Measure
MMMU-Pro	Image	558	1,172	Answer accuracy
MMAU	Audio	332	668	Answer accuracy
MMAR	Audio	332	668	Answer accuracy
Video-MMMU	Video	98	202	Answer accuracy
VideoMME-v2	Video	1,003	2,197	Answer accuracy
JointAVBench	Omni	905	1,948	Answer accuracy
OmniVideoBench	Omni	332	668	Answer accuracy
SWE-bench Multimodal	Code	100	480	Resolved-instance rate

Development feedback and final evaluation. Harbor-compatible task packages separate research execution from verification. Development feedback provides candidate scores, permitted examples, and execution diagnostics. Examples available for diagnosis are not thereby authorized as training supervision. Final test instances, reference answers, and scores remain unavailable to the research agent. At submission, weights, configuration, and loading dependencies are frozen. The verifier evaluates base and submitted models on the same fixed test instances with matched preprocessing, inference, and scoring procedures.

Research configurations. The six configurations pair GPT-5.6-sol and GPT-5.6-terra with Codex, Claude Opus 5 and Claude Opus 4.8 with Claude Code, Gemini 3.8 Flash with Gemini CLI, and Qwen 3.8 Omni Flash with Qwen Code.

Scoring and aggregation. Let $\widehat{s}_{a,b}$ be configuration a's mean target score after applying the per-loop audit rule in Appendix C, and let s_b^0 be the base score. For the eight tasks $\mathcal{B}$, the mean task score and mean change are

$$A_a = \frac{100}{|\mathcal{B}|} \sum_{b \in \mathcal{B}} \widehat{s}_{a,b}, \qquad \overline{\Delta}_a = \frac{100}{|\mathcal{B}|} \sum_{b \in \mathcal{B}} (\widehat{s}_{a,b} - s_b^0). \tag{4}$$

Tasks receive equal weight, and differences are calculated before rounding. SWE-bench Multimodal uses a 480-item test set with a base resolved-instance rate of 2.29%. Its three-loop means are included in the eight-task aggregate. Non-target evaluation and candidate analysis use one selected loop from the three for each model–benchmark pair. These analyses and auxiliary comparisons are separate from the target means.

B Additional Experimental Results

B.1 Non-target Evaluation

Each target is assigned two fixed probes to test retention outside its target category (Table 8). Image targets are checked on audio and joint audio-video understanding; audio targets are checked on image and joint audio-video understanding; video and omni targets are checked on image and audio tasks. SWE-bench Multimodal uses MMMU-Pro and MMAR to assess retention of image and audio understanding after multimodal code post-training. This fixed assignment supports comparisons across research configurations without selecting probes according to their observed gains.

Table 8: Non-target probe assignments.

Target domain	Probe 1	Probe 2
Image	MMAR	JointAVBench
Audio	MMMU-Pro	JointAVBench
Video	MMAR	MMMU-Pro
Omni	MMAR	MMMU-Pro
Code	MMMU-Pro	MMAR

For the submitted model $S_{a,b}$ and probe p, we compute $D_{a,b,p} = 100[H_p(S_{a,b}) - H_p(\theta_0)]$. Positive changes indicate improvement and negative changes indicate degradation. We average the two probe changes within a target and then weight all eight target tasks equally. These comparisons use one submitted model per target and configuration. The complete matrix contains 96 cells across six research configurations, eight target tasks, and two probes per target.

Table 9: Matched baselines for non-target probes. SWE-bench Multimodal probes use the same submitted checkpoints as its target evaluation.

Probe setting	Probe	Test items	Base accuracy (%)
Gemini / Qwen	MMMU-Pro	1,172	41.7235
	MMAR	668	72.9042
	JointAVBench	1,948	60.0616
SWE-bench Multimodal	MMMU-Pro	1,730	41.91
	MMAR	1,000	73.60

B.2 Resource Usage

Table 11 gives the numerical summaries behind the resource figure. Time and token usage describe the recorded research runs; they are considered alongside final accuracy. Agent tokens measure interaction workload rather than GPU training cost or monetary expenditure. These summaries do not constitute a matched experiment that varies only the resource budget.

C Integrity Analysis

Audits examine training-data contamination, unauthorized workspace or data access, prohibited split use, and misuse of evaluation information. Records distinguish material available to an agent, confirmed access, and evidence of prohibited use. Automated flags identify cases for review.

Coverage. Table 12 reports completed checks and recorded flags. All four audit categories cover all eight runs for each of the six research configurations. Ten runs were flagged.

Per-loop adjustment. For submitted model $S_{a,b,r}$ in loop r, audit adjustment precedes averaging:

$$\widetilde{s}_{a,b,r} = \begin{cases} s_b^0, & \text{if loop } r \text{ triggers an audit reset,} \\ H_b(S_{a,b,r}), & \text{otherwise,} \end{cases} \qquad \widehat{s}_{a,b} = \frac{1}{3}\sum_{r=1}^{3} \widetilde{s}_{a,b,r}. \tag{5}$$

Table 10: Complete non-target accuracy changes (pp). Columns follow the research configurations in the main table. Matched probe baselines are listed in Table 9.

Target	Probe	Claude Opus 5	GPT-5.6 sol	Qwen 3.8 Omni Flash	GPT-5.6 terra	Gemini 3.8 Flash	Claude Opus 4.8
MMMU-Pro	MMAR	+1.25	+1.70	−9.88	+0.50	−0.60	−8.03
	JointAVBench	+6.90	−0.03	+1.08	−1.52	+6.16	−8.86
MMAU	MMMU-Pro	−2.66	−1.98	−2.65	−0.53	−1.88	+1.26
	JointAVBench	+6.08	−0.14	+2.72	+3.05	−0.98	+8.23
MMAR	MMMU-Pro	−0.53	−0.78	+0.34	−0.44	−0.09	−1.13
	JointAVBench	−1.42	−0.80	+0.51	−0.70	−0.21	+3.82
Video-MMMU	MMAR	+1.10	+2.45	−7.04	+0.80	+0.30	−9.38
	MMMU-Pro	+1.26	−0.44	+4.69	−0.02	+3.41	+3.48
VideoMME-v2	MMAR	+1.40	+0.50	−2.40	+0.20	−0.45	−0.99
	MMMU-Pro	+0.24	+1.01	−0.60	−1.04	−1.71	+0.84
JointAVBench	MMAR	−0.99	+2.15	−4.34	+0.95	−0.75	−8.93
	MMMU-Pro	−0.70	−0.36	−2.39	−0.27	−29.86	−1.55
OmniVideoBench	MMAR	+0.65	+1.70	+0.60	+0.50	+0.00	−12.07
	MMMU-Pro	−1.21	−0.70	−0.77	+1.18	+0.00	−11.28
SWE-bench	MMMU-Pro	+1.67	+0.23	+1.24	+0.23	+0.68	+0.12
Multimodal	MMAR	−1.00	−0.80	−0.24	−0.70	−0.20	−0.74

Table 11: Recorded resource summaries. Time is in hours and agent tokens in millions.

Research model	Runtime	Mean time	Mean tokens
GPT-5.6-sol	Codex	21.7	103.9
Claude Opus 5	Claude Code	23.1	71.8
Qwen 3.8 Omni Flash	Qwen Code	17.8	69.0
GPT-5.6-terra	Codex	23.7	41.5
Gemini 3.8 Flash	Gemini CLI	17.4	35.5
Claude Opus 4.8	Claude Code	9.1	23.0

Table 12: **Recorded integrity flags and audit coverage.** Counts cover flags and completed checks.

Research model	Flagged	Trajectory	Contamination	Workspace	Split
GPT-5.6-sol	2	8/8	8/8	8/8	8/8
Claude Opus 5	2	8/8	8/8	8/8	8/8
GPT-5.6-terra	2	8/8	8/8	8/8	8/8
Claude Opus 4.8	1	8/8	8/8	8/8	8/8
Gemini 3.8 Flash	2	8/8	8/8	8/8	8/8
Qwen 3.8 Omni Flash	1	8/8	8/8	8/8	8/8

Only the flagged loop is reset; the resulting three-loop mean need not equal the base score. Table 13 pairs the original flagged-run scores with their replacements and updated means. The Gemini MMMU-Pro and JointAVBench flags correspond to reported adjusted means of 41.88% and 64.91%; the Qwen JointAVBench flag corresponds to 63.90%. Non-target tables retain raw submitted-model comparisons against matched probe baselines.

D MMResearch Implementation

MMResearch coordinates experiments within existing code-agent runtimes while retaining their native tools. It supplies research instructions and memory, validates round artifacts, and manages delivery. Debate combines Researcher, Engineer, Critic, and Artifact Analyst reviews, using independent Claude Code/Codex subagents or sequential reviews in other adapters. The procedures below expand Figure 3.

Table 13: Audit-triggering records and updated target means (%). Raw and base scores refer to the flagged historical run; the last column averages three runs after per-loop audit adjustment.

Research model	Target	Raw run	Base reset	Three-run mean
GPT-5.6-sol	MMMU-Pro	41.38	41.91	41.88
GPT-5.6-sol	VideoMME-v2	24.35	24.81	24.35
Claude Opus 5	MMAU	50.45	48.60	50.25
Claude Opus 5	OmniVideoBench	42.38	39.61	42.40
GPT-5.6-terra	MMAR	73.05	72.40	73.00
GPT-5.6-terra	JointAVBench	67.92	59.94	59.97
Claude Opus 4.8	JointAVBench	65.20	59.94	58.84
Gemini 3.8 Flash	MMMU-Pro	42.01	41.91	41.88
Gemini 3.8 Flash	JointAVBench	71.30	59.94	64.91
Qwen 3.8 Omni Flash	JointAVBench	68.42	59.94	63.90

D.1 Seven-Step Controller Procedure

The retained model $\theta^\star$ starts from the base model. Each round produces a candidate and a decision without automatically replacing it. Screening and confirmation belong to Evaluate; promotion belongs to Decide.

Algorithm 1: MMResearch seven-step research loop

Input: Task contract $\mathcal{T}$, base model θ_0, development evaluator $\mathcal{E}_{\text{dev}}$, budget B
Output: Retained model $\theta^\star$ and its loading configuration

$\theta^\star \leftarrow \theta_0$; initialize or restore checkpoint and Global/Core/Archive memory
while *remaining budget permits another round* **do**
  $m_t \leftarrow$ read Global and Core; retrieve relevant Archive records
  $d_t \leftarrow \textsc{Select}(\mathcal{T}, m_t)$
  $(e_t, h_t) \leftarrow \textsc{Diagnose}(d_t)$ ▷ anchor observations; separate hypotheses
  $p_t \leftarrow \textsc{Debate}(e_t, h_t, m_t)$
  $(\theta_t, c_t) \leftarrow \textsc{Implement}(p_t)$ ▷ save candidate and configuration
  $s_t \leftarrow \mathcal{E}_{\text{dev}}^{\text{screen}}(\theta_t)$ ▷ Evaluate: screening
  $v_t \leftarrow$ not confirmed
  if *screening supports a proposed improvement* **then**
    $v_t \leftarrow \mathcal{E}_{\text{dev}}^{\text{confirm}}(\theta_t, \theta^\star)$ ▷ Evaluate: confirmation
  end
  $r_t \leftarrow \textsc{Record}(e_t, h_t, p_t, \theta_t, c_t, s_t, v_t)$
  Index round artifacts and lessons in memory
  if *records are valid and* v_t *confirms improvement* **then**
    $\theta^\star \leftarrow \theta_t$ ▷ Decide: retain the confirmed candidate
  else
    Retain $\theta^\star$; choose refinement or a new direction ▷ Decide
  end
  Finalize the decision and checkpoint; refresh Global
end
Freeze $\theta^\star$ and its loading configuration before the budget expires
return $\theta^\star$ *and its loading configuration*

Confirmation and record validation. Screening provides inexpensive evidence for choosing which directions warrant further evaluation. Confirmation re-evaluates a proposed improvement using authorized development data held apart from the screening feedback; it is distinct from the benchmark's sealed final test. Both results are recorded with the candidate they evaluate. The controller checks for the required round artifacts and confirmation evidence before accepting a promotion record. A missing record can trigger a bounded repair request; an execution failure is recorded as a failure and does not supply a candidate score. Candidate retention follows development evidence throughout the loop.

D.2 Memory Organization and Record Structure

The three memory tiers separate a compact working summary from stable task knowledge and detailed research history. A single checkpoint stores the dynamic run state, including the current round and retained model. Global summarizes this checkpoint; a Core entry can point to it without maintaining a second, independently edited copy.

Store	Stored information	Read and write points
Global	Current plan, retained-model summary, and critical lessons.	Read at startup and SELECT; refresh after the round decision.
Core	Task rules, evaluator interface, environment facts, and reusable configurations.	Read the index at SELECT; load relevant values and add stable knowledge when established.
Archive	Hypotheses, round outcomes, failures, detailed lessons, and media references.	Retrieve relevant entries during research; index new evidence and outcomes at RECORD.
Checkpoint	Authoritative round, phase, retained candidate, and progress state.	Restore on resume; update after DECIDE and at finalization.