YOU DON'T NEED TO READ PAPERS ANYMORE: A PROVENANCE-DISCIPLINED ADVISORY TIER FOR AUTONOMOUS RESEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Autonomous research agents that work alone fail in a characteristic way: one context holds every result and the story built on it, so each new result is read through that story and the agent checks its own hypotheses against its own framing of the evidence. In practice the agent clings to its first direction until outside feedback, usually a human's, arrives: it takes run after run to leave an approach that is failing, and keeps working around problems whose solution the literature already holds. We propose separating execution from epistemic control by layering knowledge by generality: a task tier that runs the experiments, a domain-supervisor tier that owns the domain literature, and an advisory tier that answers only from general ML knowledge. YoDA (**Yo**u **D**on't need to read papers **A**nymore) implements the advisory tier as an agent outside the experiment loop. It shares neither context nor success criterion with its asker, receives only questions the supervisor has abstracted to general deep learning, answers each in a disposable per-thread agent, and marks every claim as corpus-grounded or not. Its substrate is a literature knowledge base consulted rather than browsed: 769 papers digested at registration into fixed-schema notes whose quantitative claims carry fixed provenance, under a contamination rule that never admits an asker's claim as evidence and a reading-priority score that corpus growth cannot move. On MLE-bench (22 competitions, private grading, one base model in every tier), the three-tier pipeline took 17 of 22 medals within one hour, the most of five pipelines, and 19 by 24 hours with each cell stopped at its first privately graded medal; the cell-level records show the supervisor tier producing the largest single gains and catching defects the executor could not see, and the executor overturning the supervisor's wrong priors by measurement. Code and knowledge base: `https://anonymous.4open.science/r/YoDA-06C8`.

1 INTRODUCTION

LLM agents can now run the loop of machine-learning research autonomously (propose a change, write the code, run it, read the number, decide what to try next) in end-to-end pipelines that produce paper-shaped artifacts (Lu et al., 2024; Yamada et al., 2025; Schmidgall et al., 2025), in code-space search agents that outperform most human participants on public benchmarks (Jiang et al., 2025; Chan et al., 2024), and in standardized experiment environments (Huang et al., 2023; Nathani et al., 2025).

The same period established the limits. In the most controlled expert comparison we know of, agents earn four times the expert score at a two-hour budget and fall to half of it at thirty-two hours (Wijk et al., 2024); most gains in experiment environments come from hyperparameter tuning rather than hypothesis formation (Nathani et al., 2025); and fabricated claims were found in 59% of agent-authored manuscripts accepted by automated reviewers (Gaddipati et al., 2026). An audit of 21 systems finds that collapses at three points (objective, validation and acceptance, where an internal proxy quietly substitutes for an external judgment) recur together, and reassigns the target to "autonomous execution under non-autonomous epistemic control" (Wang et al., 2026). Execution automation has advanced faster than epistemic control, and the bottleneck is now the latter.

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.

Our observation is that these failures are a matter of *position* before they are a matter of capability. An executing agent working alone keeps its attempts, its numbers and its own interpretations of them in one context, and each next interpretation is anchored to that narrative: new results are read in the direction of the standing hypothesis, the hypothesis space narrows to the frames already held, and validation happens inside the same context: validation collapse (Wang et al., 2026). A larger model delays this collapse but does not prevent it, and an agentic pipeline of two or more models is no exception: as long as its agents share one context, the validator inherits the executor's narrative.

We therefore place epistemic control *outside* the executing agent, by layering knowledge by generality into three tiers that mirror a research lab (Figure 2): the executor, the domain supervisor and the advisor (YoDA). Graduate students run the experiments and keep the lab notebook: this is the executor, at the task tier. A postdoc knows the domain literature in depth and directs their experiments: this is the domain supervisor. The advising professor knows the field broadly and hears only the questions the postdoc has distilled to general deep learning: this is YoDA, the advisory tier. For each such question YoDA calls in an outside expert, a disposable per-thread agent that answers from the lab's shelf of digested papers and leaves without keeping notes.

The analogy carries the three asymmetries of the advisory boundary. **Context**: the professor never reads the notebook, only the question written up for it (a serialized question). **Objective**: the professor is not graded on this project's metric, so the advice is task-independent methodology: alternative hypotheses, discriminating experiments, statistical design. **Grounding**: the advice comes from a literature corpus with declared provenance and marks, claim by claim, what falls outside it. Position alone is not enough, however. An advisor that answers from its own memory, or from retrieval (RAG) over external sources it cannot vouch for, gives advice nobody can check. YoDA answers instead from a curated knowledge base (KB) of its field, and that KB is what turns a general-purpose model into an advisor for the field. What makes an *agent's* advice trustworthy is then *provenance discipline*: every value in the KB has one fixed source, and an asker's claim is never evidence.

Our contributions are four.

- **Three-tier separation against context contamination.** Knowledge is layered by generality (task, domain, general), only abstracted questions cross upward, and each consultation is quarantined in a disposable per-thread agent, so neither the executor's narrative nor consultation content reaches the advisor (Section 3).
- **Provenance discipline that makes an agent's advice auditable.** Every claim is labeled as grounded in the corpus or not, every corpus value has one fixed source, and an asker's claim is never evidence, so the asker can check the advice and consultations cannot contaminate the corpus; in live use it caught an asker misattributing its own result before any GPU was spent (Sections 3.1 and 4).
- **A knowledge base that is consulted rather than browsed.** 769 papers are digested once, at registration, into fixed-schema notes, so what a consultation reads scales with the question rather than the corpus (48k–70k tokens per thread); agents maintain it, it grows where ungrounded labels reveal gaps, and its reading-priority score does not move as it grows (Section 4).
- **Outcome-level performance on MLE-bench.** On MLE-bench Low (22 competitions, private grading, one base model everywhere) the three-tier pipeline took 17 of 22 medals within one hour, the most of five pipelines (15–16 for a single agent, two tiers, a merged agent and AIBuildAI), and 19 by 24 hours with cells stopped at their first privately graded medal; transcripts locate where the supervisor tier produced a cell's largest gains (Section 5).

The margin is within single-seed noise, the three-tier column ran under an earlier protocol, and its 24-hour stage used a stopping rule the MLE-bench leaderboard counts as test-set feedback (Section 5); the quality of the advice itself is the subject of the evaluation plan (Section 6).

2 Related work

Autonomous research agents. End-to-end systems (Lu et al., 2024; Yamada et al., 2025; Schmidgall et al., 2025; Jiang et al., 2025) and evaluation infrastructure (Chan et al., 2024; Huang

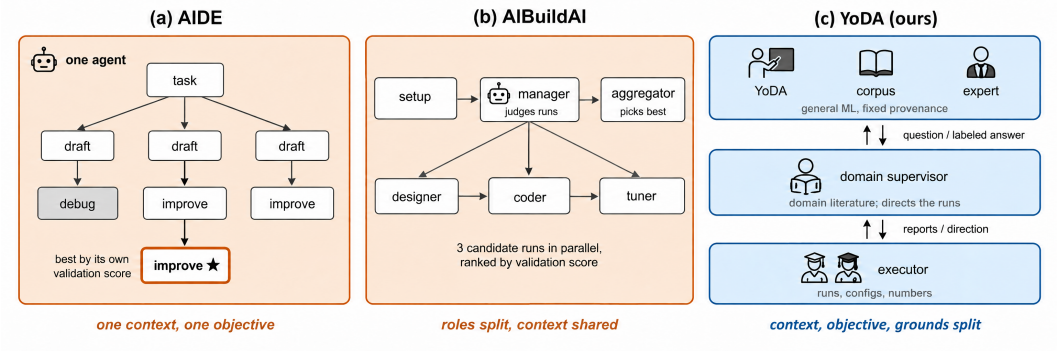


Figure 1: Where epistemic control sits. (a) AIDE: one agent drafts, debugs and improves a tree of solutions and keeps the node its own validation score ranks best. (b) AIBuildAI: a manager calls designer, coder and tuner agents over three parallel candidate runs and judges them, and an aggregator picks the submission; all roles share the task context and its score. (c) YoDA: the tiers keep separate contexts and objectives (Figure 2).

et al., 2023; Nathani et al., 2025; Wijk et al., 2024) established that the loop runs; the critical literature (Gaddipati et al., 2026; Wang et al., 2026) locates the remaining bottleneck in epistemic control (both are recent preprints; we rely on their structural diagnosis rather than their figures). Role-structured multi-agent frameworks (Hong et al., 2024; Wu et al., 2023) and research systems with reviewer and ranker agents (Gottweis et al., 2025) also separate roles, but keep one memory and one objective; our tiers separate knowledge by generality, and what may cross a boundary is defined by abstraction level and provenance, not by task assignment. We read the "non-autonomous" control of Wang et al. (2026) as control that does not reduce to the execution's internal proxies rather than as a requirement for a human: a different agent that shares neither context, objective nor grounds with the execution satisfies it, provided that agent can be trusted. Figure 1 contrasts it with AIDE and AIBuildAI, the two pipelines of Section 5, where epistemic control sits inside the loop.

Literature-grounded hypothesis generation. Closest in purpose are systems that generate hypotheses from the literature: SciAgents reasons over an ontological knowledge graph with multiple agents (Ghafarollahi & Buehler, 2024), ResearchAgent refines problems, methods and experiment designs over an academic graph with reviewing agents (Baek et al., 2025), and Zhou et al. (2024) update hypotheses iteratively against labeled examples. These systems produce hypotheses for a research task; YoDA advises a running experiment loop from outside it, and its concern is the status of each claim (what the corpus vouches for and what it does not) rather than novelty. The two are complementary: a hypothesis generator can sit at the domain tier and consult the advisor like any other asker.

Debate, critics, and shared priors. Buying reliability with independent agents is not new: debate as an oversight mechanism (Irving et al., 2018), multi-agent debate for factuality (Du et al., 2023), persuasive debaters leading to truthful answers (Khan et al., 2024), AI and tool-interactive critics (Bai et al., 2022; Gou et al., 2024; McAleese et al., 2024), and the limits of self-critique where no external information enters (Madaan et al., 2023; Huang et al., 2024), which the sycophancy literature predicts from inside a shared frame (Sharma et al., 2023). Debaters split by stance over shared grounds and are adjudicated by persuasion or a judge; YoDA splits the grounds themselves (the asker holds the domain and its data, the advisor a provenance-fixed corpus), and disagreement is adjudicated by auditable grounding. What makes the critic external is position, not a role label or a trained head. Position does not separate priors, though: every tier runs the same class of base model, so the ungrounded part of an answer is where correlated error between asker and advisor can live. The labels bound that exposure, and a cross-model advisor is part of the evaluation plan.

Retrieval, memory, and literature tools. RAG (Lewis et al., 2020) and GraphRAG (Edge et al., 2024) take the corpus as given, and agent memories (Packer et al., 2023; Shinn et al., 2023) maintain their stores for the agent's own continuity. Our problem is the write path: who builds the representation being retrieved from, under which per-field provenance, and how it is defended against its

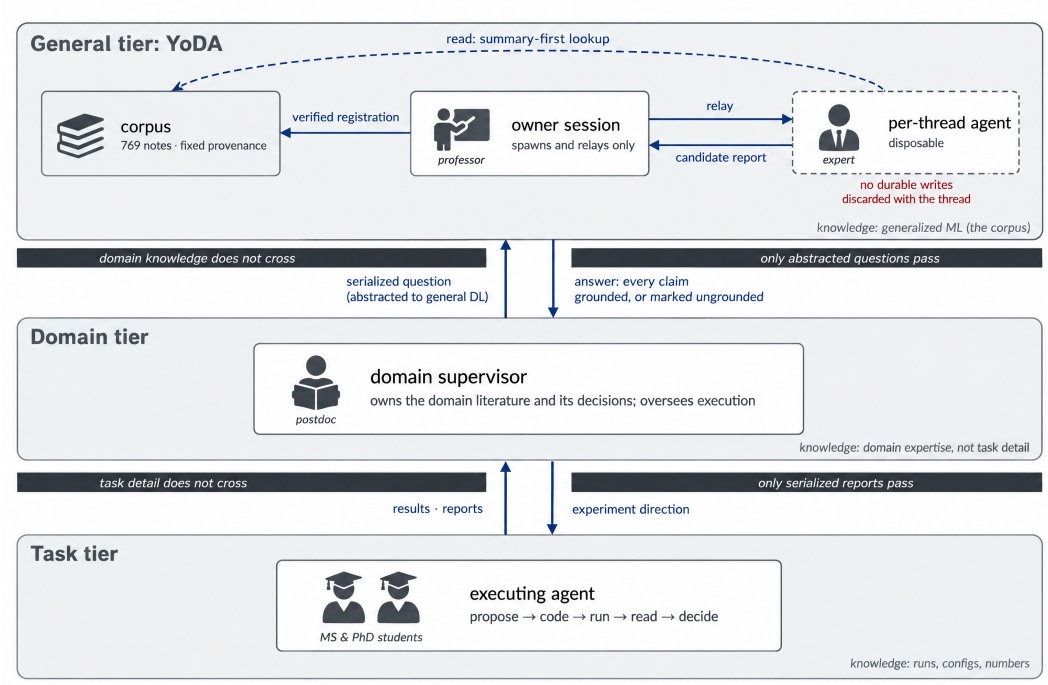


Figure 2: The three tiers and the quarantine, drawn with the lab analogy of Section 1. Knowledge is pinned to each tier's level of generality (thick boundaries); only serialized, abstracted questions and labeled answers cross upward, and a consultation lives in a disposable per-thread agent whose one durable channel is a verified candidate registration.

own consultation traffic. A note also differs from a chunk: it is secondary literature that organizes judgments about a paper into a fixed schema. The nearest neighbors, citation-grounded literature-QA agents (Skarlinski et al., 2024; Asai et al., 2024), synthesize with cited support at answer time; YoDA maintains a curated corpus consumed by agents under contamination rules, with labels that separate corpus-vouched from parametric content, and its score is a weight for what to read next rather than a bibliometric verdict of influence.

3 System Architecture: Separation (Hierarchy and Quarantine)

The deployment is layered by the generality of knowledge (Figure 2). The *task tier* is the executor: its knowledge is the task itself (run histories, configurations, numbers), and its files are the deployment's only memory, so every report or question going up carries the history slice it depends on. The *domain tier* is a supervisor that collects the domain's literature, owns every decision requiring domain expertise, and oversees execution. The *general tier* is YoDA, answering from a corpus of generalized ML knowledge. Knowledge is pinned to its level: task detail stays at the task tier, domain knowledge never crosses into the advisor, and the only thing that crosses a boundary upward is a question abstracted to the next level. The layering is also a division of depth: the general tier holds hundreds of papers as digests anchored to their abstracts, while the domain tier reads a few dozen papers whole. An advisory answer is therefore a map (levers, applicability conditions, caveats, pointers) that the supervisor deepens by reading the pointed-to papers before it directs any experiment.

The unit of work at the advisory boundary is a consultation: the supervisor serializes a stuck point (an unexplained result, competing hypotheses, the allocation of an experiment budget) into a question, and the advisor returns a grounded answer and discriminating experiments. Each asymmetry does work here.

Context asymmetry is enforced by the interface. The advisor cannot read the asker's context, so the asker must serialize its state, and the serialization is itself the first intervention: numbers are separated from their narrative, and interpretations become explicit claims that can be contested. In the trace of Section 3.1, the asker's misattribution was caught not because the advisor was smarter, but because the serialized claim could be checked against note content without the narrative attached.

Objective asymmetry sets the level of the advice. Nothing in the advisor's context ties it to the asker's metric moving, so its advice comes not as task-specific answers ("use this value") but as methodology: alternative hypotheses, the design and cost-ordering of discriminating experiments, statistical design, pre-registration. The second correction of Section 3.1 (a single-seed comparison cannot reach significance) came from this level, not from the corpus.

Grounding asymmetry changes the direction of trust. Because answers are marked claim by claim as grounded or ungrounded, the asker receives advice not as authority but as something it can audit: only by knowing which sentence came from the corpus can it ask whether the corpus, the advisor's parametric memory, or its own data is wrong. That the corrections of Section 3.1 ran in both directions is a consequence of this structure.

Quarantine. Two actors implement the general tier: the *owner*, the long-lived session that holds the corpus and registration authority, and disposable *per-thread agents* it spawns, one per consultation thread. The owner only spawns and relays. The thread agent takes the thread whole, is the single point task-specific content ever touches, can persist nothing (Section 4.3), and is discarded with the thread. What returns to the owner is administrative (whom the answer went to, which candidates were reported), never the answer's content, so consultation detail does not accumulate in the advisor and cannot become a narrative it anchors on. Separation thus blocks contamination at three boundaries (task detail hardening into domain knowledge, domain knowledge leaking into the general corpus, consultation content settling into the advisor), each by leaving one narrow interface (the serialized question; the verified candidate report) and removing every other path.

3.1 Example from operation: a consultation with a domain supervisor

The protocol's first live user was a domain supervisor overseeing autonomous pose-estimation experiments in the same project; this is a deployment record, not a controlled evaluation, and the full trace is in Appendix D. Only what the tier interface admits crossed the boundary. Going up was a serialized question at the level of general deep learning (why fine-tuning a ViT lands in one of two outcomes), and in the next round the supervisor's own diagnostic result with the reading it favored. Coming down was an answer whose literal mechanism claim was marked ungrounded, grounded alternative explanations with a cost-ordered list of discriminating experiments, and then a rebuttal from note content: the diagnostic weakened the supervisor's favored hypothesis rather than supporting it, because the paper behind that hypothesis predicts preserved training loss (Kumar et al., 2022). The supervisor re-ranked its hypotheses before spending GPU time, and when the advisor showed that its planned single-seed comparison could not reach significance at any level, it pre-registered a multi-seed design with fixed decision rules.

Corrections also ran the other way: the supervisor's logs refuted one of the advisor's concerns, and an ungrounded sketch the advisor had offered turned out partly wrong and was corrected once the relevant note existed. The gaps the answers exposed went to the owner as candidate reports, were verified and registered, and the claims they supported moved from ungrounded to grounded. The labels did not hide the advisor's errors; they marked in advance where errors could live.

4 System Architecture: The advisor's substrate (corpus and discipline)

The current instance's substrate is the deep-learning model-optimization literature: 769 papers across 15 topics (quantization, pruning, distillation, efficient attention, serving, scaling laws, and so on) with the boundary drawn by advisory need, so the training-dynamics and representation-learning papers that consultations reach for are in scope. The pipeline is Python standard library only; search and scoring run offline, and only metadata collection (titles, abstracts, citation records) touches the network.

Table 1: Provenance per corpus field. The metadata rows are pipeline-enforced; the synthesis row is schema-marked and sample-audited (Appendix C); hand-written notes bypassing the path are forbidden. Citations are enumerated through the Semantic Scholar Graph API

Value	Only admissible source
Title	arXiv metadata, fetched by id
Citation totals and per-year histogram	exhaustive enumeration via the Semantic Scholar Graph API
Result figures, quantitative claims in notes	the authors' abstract, never beyond it
Synthesis sections of a note	the note-writing agent, from title and abstract plus its general knowledge; marked as synthesis by the schema
Scores and tiers	pipeline recomputation (no manual entry)
An asker's claims	not admissible: grounds for a candidate report only

4.1 Fixed-schema notes: a digest, not an extract

One paper is one note, and every note carries the same seven sections in the same order: Problem, Key idea, Method details, Results, Limitations and criticism, Practical takeaways, Related work. All 769 notes share exactly one heading sequence (Appendix C.2). A consumer reads sections rather than files, and comparison across notes is structurally well founded. A note is a digest, not an extract, and the trust story depends on saying so precisely. Its quantitative content (any result figure, any experimental claim) may come only from the authors' abstract; Problem and Key idea restate the abstract; the remaining sections are the note-writer's structured synthesis around that fixed core, and the schema keeps the distinction inspectable. A grounded label therefore certifies one hop: *the corpus's digest supports this claim*. For a claim resting on a synthesis section that is the note-writer's reading of the abstract, not the paper's own statement, and the digest-to-paper hop is a sampled property, audited rather than guaranteed (Section 6). No metric is ever written into a note: citation counts, scores and tiers live in one index per topic, from which every published document is regenerated.

4.2 Consumption and grounding labels

Consumption is tiered by token budget. The search tool returns roughly 250-byte summaries per paper, and a note (about 5.5 KB) is opened only when more is needed and read one section at a time, so what a consultation reads scales with the question rather than with the corpus (a single-session thread cost 48k–70k tokens; Appendix D). Two rules govern answers. First, every claim bears a grounding label: tied to the supporting note, or explicitly marked as resting on general knowledge outside the corpus. Omission is not permitted; mixing both labels in one answer is normal, and where the ungrounded label recurs maps the corpus's gaps and feeds registration candidates. Second, length follows content: rhetoric and repetition may be cut, findings, caveats and configuration values may not.

4.3 Maintenance by agents: the contamination rule

The corpus is maintained by the same kind of agent that consumes it, and the central constraint is *contamination prevention*: every value entering the corpus has a fixed provenance (Table 1), and claims heard during consultation cannot be that provenance. If an asker says "this paper showed X," that sentence is a *reason* to register the paper but never a *license* to write X into its note; the note may state X as the paper's finding only if the authors' abstract says it. Registration is two-staged: an agent that needs a paper the corpus lacks cannot register it; it sends a fixed-format candidate report (arXiv id, title, topic, role, a one-line reason; Appendix B.1) to the owner, which verifies the id against arXiv, judges topic and scope, and only then enqueues the paper. The write rule for subagents is one sentence: *write nothing except the file your brief commissions*. A subagent dies

Agent	LLM	h	G	S	B	Any %
YoDA (ours)	Opus 5	1	11	5	1	77.3
YoDA (ours)	Opus 5	24	11	5	3	86.4
AIDE	Opus 5	1	12	3	1	72.7
AIBuildAI	Opus 5	1	11	4	1	72.7
MLE-bench leaderboard						
Famou-Agent 2.0	Gemini-3-Pro	24	11.0	4.0	2.7	80.3
MLEvolve	Gemini-3-Pro	12	10.3	5.3	2.0	80.3
PiEvolve	Gemini-3-Pro	24	11.0	4.7	2.0	80.3
MARS+	Gemini-3-Pro	24	11.0	3.7	2.7	78.8
AIBuildAI	Opus 4.6	24	8.3	5.7	3.0	77.3
Famou-Agent 2.0	Gemini-2.5-Pro	24	9.0	4.3	3.3	75.8
ML-Master 2.0	DeepSeek-V3.2	24	6.7	7.0	3.0	75.8
MARS	Gemini-3-Pro	24	11.7	3.3	1.3	74.2

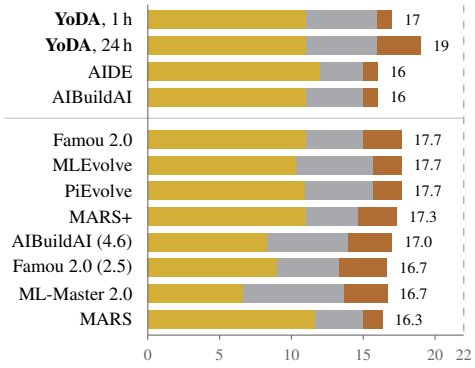


Figure 3: MLE-bench Low (22 competitions): gold, silver and bronze. Top: our runs on Claude Opus 5 (the YoDA rows come from earlier-protocol arms, see text). Bottom: the leading entries of the public MLE-bench leaderboard at their reported budgets, with medals recomputed from their published grading reports.

with its session while files persist, so an uncommissioned write lets unreviewed conclusions outlive the reasoning that produced them; what persists is decided by the owner, after looking.

4.4 A READING-PRIORITY SCORE INDEPENDENT OF CORPUS COMPOSITION

Each paper carries a recency–citation score

$$s \;=\; 100\left(0.7 \cdot \frac{\log(1+v)}{\log(1+V)} \;+\; 0.3 \cdot m\right), \qquad V = 10^4, \tag{1}$$

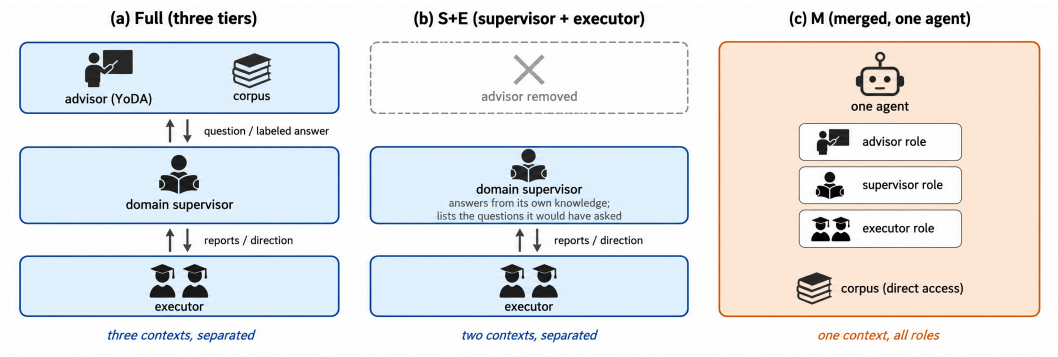
where v is the annualized citation velocity over the last three calendar years and m the fraction of the paper's citations that arrived in that window; tiers are absolute cuts on s (S $\geq$ 75, A $\geq$ 66, B $\geq$ 56, C $\geq$ 46, D below). Both constants are fixed rather than corpus statistics: V was chosen once at design time and frozen, and the cuts are not percentiles. An earlier version cut tiers at within-corpus percentiles, and registering one paper moved the tiers of papers nobody had touched. Now a score reads nothing but the scored paper's own citation record, so corpus growth cannot move existing entries, and a regression test pins the property. The score is a reading-priority weight, not a quality verdict.

5 EXPERIMENTS

We evaluate on the Low split of MLE-bench (Chan et al., 2024): 22 Kaggle competitions that the benchmark re-splits into public and private parts. A cell is one competition under one pipeline with a fixed wall-clock budget; an external clock grades the deliverable privately at each full hour, and no grade, not even "medal or not", ever reaches an agent. The one exception is the 24-hour stage of the earlier arms, run for YoDA on the five competitions still without a medal after three hours: it stopped a cell at its first medal, and since the agents knew the rule, every further hour told them they were still below bronze. The MLE-bench leaderboard counts such runs as test-set feedback, so we report that stage separately. Every agent runs Claude Opus 5 on the public files only, one seed per cell. In accordance with the benchmark criteria, we used two sessions in which the computational capacity of the RTX 4090 was capped at half, matching that of the A10. Appendix A gives the protocol, the per-competition grades.

5.1 RESULTS

Figure 3 compares YoDA with an **AIDE** , a single agent running AIDE's draft–debug–improve tree search (Jiang et al., 2025), and **AIBuildAI**, an external multi-stage pipeline that stops on its own judgment. Within one hour YoDA takes the most medals, 17 of 22 against 16 for each, and the most silver; the AIDE takes one more gold. By 24 hours, YoDA reaches 19 medals. For scale, the figure also lists the leading entries of the public MLE-bench leaderboard on the same split (Chan et al.,



Pipeline	G	S	B	Total
Full (3 tiers)	11	5	3	**19**
M (merged)	11	3	2	16
S+E (2 tiers)	9	4	2	15

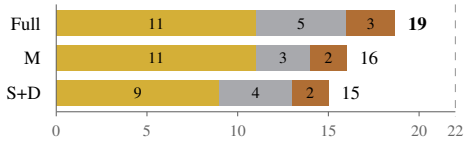


Figure 4: Ablation of YoDA's structure at 24hr, same 22 competitions. S+E removes the advisor; M merges the three roles and their contexts into one agent with corpus access.

2024), with gold, silver and bronze recomputed from their published grading reports. With other base models, 12–24-hour budgets and three seeds each, they average 16.3 to 17.7 medals (74–80%), AIBuildAI on Claude Opus 4.6 among them at 17.0 over 24 hours. YoDA's one-hour run falls inside that band. The domain-specific knowledge dissemination system demonstrated rapid convergence.

5.2 Ablation Study

Figure 4 removes one part at a time, in the fixed-budget arm. **S+E** (Supervisor & Executor) removes the advisor: the supervisor answers from its own knowledge and lists the questions it would have sent. **M** (Merged 3 Layers) keeps the roles and the corpus but not the separation: one agent holds every role and its context. The full hierarchy takes 17 medals, M 16 and S+E 15 (14 of 21 without the one cell whose tiers never connected).

Timing, not content, limited the supervisor. Transcripts, rather than the agents' own reports, show what the tiers did. The supervisor's first message almost always arrived in time: in all 22 S+E cells it was sent before the launch mark, and in 20 it reached the executor before the first training run. Later messages were the problem. An agent sees new messages only between its turns, so an executor busy with a long run received later directions in one batch, in several cells only after its deliverable was final. In nine cells both agents stopped to wait for each other while a job ran, and every restart came from an outside notice that stated only the clock time.

Direction that arrived in time mattered, and corrections ran both ways. Where a direction landed in time, it produced the largest gain of its cell by the executors' own measurements: reading an ImageNet classifier's dog classes without any training on dogbreed, a higher input resolution on histopathology and melanoma, and on taxi the catch that validation rows had been cleaned while the graded test set had not. Supervisors also caught defects their executors could not see, such as a blend script that would have dropped every model at the deadline. In the other direction, executors disproved about half of the supervisors' domain assumptions by measurement.

Methodology stayed out of reach. What the two tiers could not settle was methodology. The questions the supervisors saved for the absent advisor converge on how to select models without overfitting the selection set, which calibration assumptions hold, which held-out test to trust, and whether a score pooled over folds can be compared with a single-fold one. These are the questions this paper assigns to the advisor, and no measurement inside a cell can answer them.

6 LIMITATIONS AND EVALUATION PLAN

The evidence comes from one setting. Everything reported comes from one knowledge base (the model-optimization literature), one project and one agent harness. Live use tests the supervisor–advisor boundary and MLE-bench tests the executor–supervisor boundary, but neither is yet a controlled measurement of advice quality, and the instruction and message-delivery behavior we report may differ on another harness. **Depth stops at the abstract, by design.** Notes take numbers only from the authors' abstracts, which keeps every value traceable but limits how deep a note goes. Full papers are read at the domain tier, yet the limit still shows in the advisor's own answers: questions about effect sizes systematically come back ungrounded, while claims about mechanisms, applicability and caveats stay grounded. Reading is not removed but moved to registration, so the advisor can vouch for no more than the quality of that digest. **Note quality is audited, not guaranteed.** The pipeline enforces provenance only for metadata; whether a note's summary is faithful to its abstract rests on writing rules and sampled checks rather than a formal guarantee.

The evaluation plan targets these gaps: label attribution fidelity with AIS and ALCE (Rashkin et al., 2021; Gao et al., 2023); an isolation ablation measured with the sycophancy protocol of Sharma et al. (2023); tier and label ablations; token cost against chunk RAG and full-text reading; answer quality on LitQA2 and ScholarQABench (Skarlinski et al., 2024; Asai et al., 2024); a cross-model advisor; and outcome tests on MLAgentBench, MLGym and RE-Bench (Huang et al., 2023; Nathani et al., 2025; Wijk et al., 2024), whose long budgets test the thesis directly.

7 CONCLUSION

If the bottleneck of autonomous research is epistemic control rather than execution, control should not be improved inside the execution but moved outside it. YoDA layers knowledge by generality and implements control at the top of that hierarchy as an independent advisor agent: sharing neither context, objective nor grounds with the execution, answering from a provenance-disciplined literature corpus, marking the grounding of every claim, and advising at the level of task-independent methodology. Under these disciplines a corpus can be maintained by the same kind of agent that consumes it without the consultation loop contaminating the record, and the gaps an answer exposes become, after verification, the grounds of the next answer. We watched this design change a supervisor's experiment before GPU was spent, and on MLE-bench we measured what a supervisor tier adds and what the channel between tiers withholds. Whether the advisor changes outcomes is the next measurement.

AI USE STATEMENT

We used generative AI tools to discuss and refine the conceptual framing and hypotheses, provide feedback on experimental methodology and analysis, implement and run the software for the system and the experiments, assist with translation, and assist with the interpretation of results. Additionally, we used generative AI tools for brainstorming, for creating figures, and for drafting and editing portions of the manuscript to improve clarity and readability. We did not use them to generate synthetic data sets or to formulate or prove mathematical claims. All research decisions, experimental design, implementation, analyses, and conclusions were made, conducted, or verified by the authors, and all AI-assisted content was reviewed by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Every published corpus figure is regenerated from a single source (the per-topic corpus index) by one entry-point script; all constants of the score definition live in the scoring code, and the published documents are rendered from them. The measurement window can be pinned to a specific month by an environment variable, so rescoring at a past point in time is reproducible. The MLE-bench tables are generated from the recorded grade files by a script, and the per-cell records (launch prompts, clocks, hourly snapshots, final reports, tier-activation timings) are kept alongside them. The corpus, pipeline, instruction files and tier templates are released as a public repository (`https://anonymous.4open.science/r/YoDA-06C8`), including the instruction texts excerpted in Appendix B; the consultation thread of Appendix D will be released as an anonymized transcript in the supplementary material.

REFERENCES

Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'arcy, David Wadden, Matt Latzke, Minyang Tian, Pan Ji, Shengyan Liu, Hao Tong, Bohao Wu, Yanyu Xiong, Luke Zettlemoyer, Graham Neubig, Dan Weld, Doug Downey, Wen-tau Yih, Pang Wei Koh, and Hannaneh Hajishirzi. Openscholar: Synthesizing scientific literature with retrieval-augmented LMs. *arXiv preprint arXiv:2411.14199*, 2024.

Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. ResearchAgent: Iterative research idea generation over scientific literature with large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics (NAACL)*, 2025.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Lilian Weng, and Aleksander Mądry. MLE-bench: Evaluating machine learning agents on machine learning engineering. *arXiv preprint arXiv:2410.07095*, 2024.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *International Conference on Computer Vision (ICCV)*, 2021.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

Sasi Kiran Gaddipati, Diyana Muhammed, Farhana Keya, Gollam Rabby, and Sören Auer. ML-Replicate: Benchmarking autonomous research systems for machine learning reproducibility. *arXiv preprint arXiv:2605.16616*, 2026.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Empirical Methods in Natural Language Processing*, 2023.

Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann LeCun. RankMe: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *International Conference on Machine Learning (ICML)*, 2023.

Alireza Ghafarollahi and Markus J. Buehler. SciAgents: Automating scientific discovery through multi-agent intelligent graph reasoning. *arXiv preprint arXiv:2409.05556*, 2024.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Petar Sirkovic, Artiom Myaskovsky, Grzegorz Glowaty, Felix Weissenberger, Alessio Orlandi, Dan Popovici, Anil Palepu, Keran Rong, Ryutaro Tanno, Khaled Saab, Fan Zhang, Jacob Blum, Andrew Carroll, Kavita Kulkarni, Nenad Tomasev, Dina Zverinski, Ivor Rendulic, Elahe Vedadi, Florian Hasler, Luka Rimanic, Marina Boia, Ivan Budiselic, Ben Feinstein, Mathias Bellaiche, Tom Sheffer, Jan Freyberg, Jeremy Ratcliff, Ottavia Bertolli, Katherine Chou, Avinatan Hassidim, Burak Gokturk, Amin Vahdat, Yuan Guan, Vikram Dhillon, Eeshit Dhaval Vaishnav, Byron Lee, Tiago R D Costa, José R Penadés, Gary Peltz, Yossi Matias, James Manyika, Demis Hassabis, Yunhan Xu, Pushmeet Kohli, Annalisa Pawlosky, Alan Karthikesalingam, and Vivek Natarajan. Accelerating scientific discovery with Co-Scientist. *arXiv preprint arXiv:2502.18864*, 2025.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, 2024.

Sirui Hong, Mingchen Zhuge, Jiaqi Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. Metagpt: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, 2024.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations*, 2024.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. MLAgentBench: Evaluating language agents on machine learning experimentation. *arXiv preprint arXiv:2310.03302*, 2023.

Geoffrey Irving, Paul Christiano, and Dario Amodei. AI safety via debate. *arXiv preprint arXiv:1805.00899*, 2018.

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 preprint arXiv:2502.13138*, 2025.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R. Bowman, Tim Rocktäschel, and Ethan Perez. Debating with more persuasive LLMs leads to more truthful answers. In *International Conference on Machine Learning (ICML)*, 2024.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations (ICLR)*, 2022.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

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 preprint arXiv:2408.06292*, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, 2023.

Nat McAleese, Rai Michael Pokorny, Juan Felipe Ceron Uribe, Evgenia Nitishinskaya, Maja Trebacz, and Jan Leike. LLM critics help catch LLM bugs. *arXiv preprint arXiv:2407.00215*, 2024.

Deepak Nathani, Lovish Madaan, Nicholas Roberts, Nikolay Bashlykov, Ajay Menon, Vincent Moens, Amar Budhiraja, Despoina Magka, Vladislav Vorotilov, Gaurav Chaurasia, Dieuwke Hupkes, Ricardo Silveira Cabral, Tatiana Shavrina, Jakob Foerster, Yoram Bachrach, William Yang Wang, and Roberta Raileanu. MLGym: A new framework and benchmark for advancing AI research agents. *arXiv preprint arXiv:2502.14499*, 2025.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. Measuring attribution in natural language generation models. *arXiv preprint arXiv:2112.12870*, 2021.

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. Agent laboratory: Using LLM agents as research assistants. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2025.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, Shauna Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. Towards understanding sycophancy in language models. *arXiv preprint arXiv:2310.13548*, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Michael D. Skarlinski, Sam Cox, Jon M. Laurent, James D. Braza, Michaela Hinks, Michael J. Hammerling, Manvitha Ponnapati, Samuel G. Rodriques, and Andrew D. White. Language agents achieve superhuman synthesis of scientific knowledge. *arXiv preprint arXiv:2409.13740*, 2024.

Shuai Wang, Xinyuan Tian, Pang-Chi Liu, and Yize Zhao. Workflow closure is not scientific closure in auto-research systems. *arXiv preprint arXiv:2605.26200*, 2026.

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 preprint arXiv:2411.15114*, 2024.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. The AI scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint arXiv:2504.08066*, 2025.

Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Josh Susskind. Stabilizing transformer training by preventing attention entropy collapse. In *International Conference on Machine Learning (ICML)*, 2023.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. Revisiting few-sample BERT fine-tuning. In *International Conference on Learning Representations (ICLR)*, 2021.

Yangqiaoyu Zhou, Haokun Liu, Tejes Srivastava, Hongyuan Mei, and Chenhao Tan. Hypothesis generation with large language models. In *Proceedings of the 1st Workshop on NLP for Science (NLP4Science)*, 2024.

A THE MLE-BENCH EXPERIMENTS IN FULL

This appendix gives the protocol and the results of the MLE-bench experiments of Section 5. Every grade is a private-split grade computed by the benchmark's own grader on a file taken from disk by an external clock. The tables are generated by scripts from the recorded grade files and from the public leaderboard.

A.1 SETUP

Benchmark and grading. MLE-bench (Chan et al., 2024), Low split: 22 Kaggle competitions, each re-split by the benchmark from the original training set into a public part and a private test part. A cell is one competition run under one condition. Its deliverable is a `submission.csv`; the benchmark's grader scores it against the private answers and assigns gold, silver or bronze by the competition's medal thresholds.[1] We report the score, the rank the score would have taken on the original leaderboard, and the medal.

Budget protocol. The conditions of the ablation arm (S+E, M and the AIDE) use a fixed wall-clock budget of one hour counted from a launch mark. An external clock, independent of every agent, copies the deliverable at each full hour and grades it privately; nothing an agent does after the mark counts, and no grade, not even the one bit "medal or not", is sent back. The base model is the same for every agent in every condition (Claude Opus 5), the data are the benchmark's public files only, the answer-key directory is off-limits, inference from the re-split structure (class marginals derivable from public row counts) is forbidden, and only generic pretrained weights may be downloaded. Two cells run concurrently on one 24 GB GPU with a stated per-cell share; an agent is spawned three to eight minutes before its mark to read its brief, and measured activity begins 97–311 s before the mark (mean 187 s over 20 cells), so the effective budget is one hour plus at most about five minutes. The full 24-hour stage also ran the all competitions.

Conditions.

- **Full (three tiers).** Executor, domain supervisor and the advisor of this paper, from the earlier arms: a 22-competition arm with a 20-minute training budget per competition and a single GPU. This protocol differs from the ablation arm's (shorter training budget, half GPU, no head start).
- **S+E (supervisor and executor; two tiers).** Executor and supervisor with the advisor removed: the supervisor answers from its own knowledge, marks unverified claims, and keeps a list of the questions it would have sent to an advisor.
- **M (merged, one agent).** A single agent that holds the roles and contexts of executor, domain supervisor and advisor at once, with direct read access to the corpus and its lookup tool; its brief differs from the executor's in three blocks (role, literature access, and self-direction with self-audit).
- **AIDE (one agent).** A single agent that runs the draft/debug/improve tree search of AIDE (Jiang et al., 2025) itself (five drafts, then a coin flip between debugging a buggy leaf and improving the best node, a journal of nodes, the best non-buggy node's file as the deliverable), with the same rules, budget and model.
- **AIBuildAI (external pipeline).** A closed multi-stage pipeline run on the same model and rules. Its sweep ran one cell at a time with the GPU to itself, before the arm's shared-GPU schedule. It terminates on its own judgment; its one-hour runs ended after 18–57 minutes (mean 34).

A.2 RESULTS AT ONE HOUR

Table 2 gives the private-split grade of every pipeline on every competition at one hour.

[1] The grader returns no medal judgment for the two text-normalization competitions; those were scored with an equivalent script, as in every arm reported here.

Table 2: Private-split grades at one hour, per competition (score, original leaderboard rank, medal G/S/B). Full = three tiers (reference protocol, see text); AIDE. = single agent; AIBuildAI = external control.

		Full (3-tier)			AIDE. (1 agent)			AIBuildAI		
#	competition	score	rank	medal	score	rank	medal	score	rank	medal
1	cactus	1.00000	1	G	1.00000	1	G	1.00000	1	G
2	aptos	0.92091	122	S	0.92211	99	S	0.93083	11	G
3	denoise	0.01096	3	G	0.00916	3	G	0.01159	4	G
4	insults	0.94842	1	G	0.95739	1	G	0.94712	1	G
5	dogbreed	0.18175	199	–	0.10789	154	–	0.21524	313	–
6	dogscats	0.00618	1	G	0.00440	1	G	0.00753	1	G
7	histo	0.99321	10	G	0.99820	9	G	0.99805	9	G
8	jigsaw	0.98820	3	G	0.98651	338	B	0.98470	1519	–
9	leaf	0.00014	25	S	0.00042	26	S	0.00097	29	S
10	birds	0.91766	14	S	0.95434	2	G	0.95796	1	G
11	taxi	3.76299	849	–	3.89374	922	–	3.86887	913	–
12	nomad	0.04859	1	G	0.05072	1	G	0.05088	1	G
13	plant	0.99658	1	G	0.99286	1	G	0.98654	1	G
14	pizza	0.69598	92	B	0.69017	102	–	0.79795	33	S
15	ranzcr	0.94683	1184	–	0.95435	1092	–	0.93668	1223	–
16	melanoma	0.92914	810	–	0.90822	1827	–	0.91785	1488	–
17	spooky	0.23756	25	S	0.21195	20	S	0.25303	34	S
18	tabdec	0.96298	1	G	0.96330	1	G	0.96340	1	G
19	tabmay	0.99777	215	–	0.99798	174	–	0.99729	242	–
20	tnen	0.99780	4	G	0.99737	10	G	0.99698	11	G
21	tnru	0.98982	11	S	0.99025	10	G	0.98079	38	B
22	whale	0.99237	3	G	0.99385	1	G	0.98883	12	S
medals			17/22			16/22			16/22	

A.3 Results at 24 hours

Table 3 lists the results of the public MLE-bench leaderboard on the Low split together with YoDA's at 24 hours.

B Prompts in full

The deployment is configured entirely by the prompt files below: the advisor tier's universal rules and owner role, the domain tier's instruction pair, the task tier's instruction file, and the launch briefs of the MLE-bench runs. They are reproduced verbatim from the public repository and the experiment records; only the typography is converted to ASCII for typesetting (dashes, arrows, bullets and box-drawing lines). Placeholders such as `<DOMAIN>`, `{COMP}` or `{EXECUTOR}` are filled when a deployment is set up or a cell is launched.

B.1 Advisor tier: the universal rules (`.claude/CLAUDE.md`)

Injected into every session that starts in the knowledge base, the owner and every per-thread agent alike. Its section "Reporting a paper the corpus lacks" fixes the candidate report format. supplement/prompts/advisor$_u niversal_r ules.txt$

Table 3: MLE-bench Low split: the public leaderboard (README of openai/mle-bench, 2026-04-24), each entry at its reported running time, and YoDA at one hour and at 24 hours. Medals are the medal rate times 22.

Agent	LLM	Time (h)	Low medal rate (%)	Medals (of 22)
MLE-bench leaderboard, main table (mean ± s.e. over seeds)				
Famou-Agent 2.0	Gemini-3-Pro-Preview	24	80.30 ± 1.52	17.7
MLEvolve	Gemini-3-Pro-Preview	12	80.30 ± 1.52	17.7
PiEvolve	Gemini-3-Pro-Preview	24	80.30 ± 1.52	17.7
CAIR MARS+	Gemini-3-Pro-Preview	24	78.79 ± 1.52	17.3
AIBuildAI	Claude-Opus-4.6	24	77.27 ± 0.00	17.0
Famou-Agent 2.0	Gemini-2.5-Pro	24	75.76 ± 1.52	16.7
ML-Master 2.0	Deepseek-V3.2-Speciale	24	75.76 ± 1.51	16.7
CAIR MARS	Gemini-3-Pro-Preview	24	74.24 ± 1.52	16.3
PiEvolve	Gemini-3-Pro-Preview	12	74.24 ± 3.03	16.3
Leeroo	Gemini-3-Pro-Preview	24	68.18 ± 2.62	15.0
CAIR MLE-STAR-Pro-1.5	Gemini-2.5-Pro	24	68.18 ± 2.62	15.0
R&D-Agent	gpt-5	12	68.18 ± 2.62	15.0
CAIR MLE-STAR-Pro-1.0	Gemini-2.5-Pro	12	66.67 ± 1.52	14.7
Thesis	gpt-5-codex	24	65.15 ± 1.52	14.3
Operand ensemble	gpt-5 (low verbosity/effort)	24	63.64 ± 0.00	14.0
Famou-Agent	Gemini-2.5-Pro	24	62.12 ± 1.52	13.7
InternAgent	deepseek-r1	12	62.12 ± 3.03	13.7
AIRA-dojo	o3	24	55.00 ± 1.47	12.1
R&D-Agent	o3 + GPT-4.1	24	51.52 ± 4.01	11.3
Neo multi-agent	undisclosed	36	48.48 ± 1.52	10.7
ML-Master	deepseek-r1	12	48.48 ± 1.52	10.7
R&D-Agent	o1-preview	24	48.18 ± 1.11	10.6
AIDE	o1-preview	24	35.91 ± 1.86	7.9
AIDE	claude-3-5-sonnet-20240620	24	19.70 ± 1.52	4.3
AIDE	gpt-4o-2024-08-06	24	18.55 ± 1.26	4.1
OpenHands	gpt-4o-2024-08-06	24	12.12 ± 1.52	2.7
AIDE	llama-3.1-405b-instruct	24	10.23 ± 1.14	2.3
MLAB	gpt-4o-2024-08-06	24	4.55 ± 0.86	1.0
MLE-bench leaderboard, runs with test-set feedback (listed apart by the leaderboard)				
Disarray	Ensemble (Claude-Opus-4.5, Claude-Sonnet-4.5, GPT-5.2-Codex, Gemini-3-Pro-Preview)	24	90.91 ± 0.00	20.0
LoongFlow	Gemini-3-Flash-Preview	24	77.27 ± 0.00	17.0
This paper: YoDA on Claude Opus 5, one seed				
YoDA (ours)	Claude Opus 5	1	77.27	17
YoDA (ours)	Claude Opus 5	24	86.36	19
S+E (supervisor and executor)	Claude Opus 5	24	68.18	15
M (merged)	Claude Opus 5	24	72.73	16

B.2 Advisor tier: the owner role (.claude/YODA.md)

Read only by the session that owns the knowledge base (the professor of the lab analogy); a per-thread agent is told at its first line not to open it. supplement/prompts/advisor$_{o}wner_{r}ole.txt$

B.3 Domain tier: the supervisor's instructions (templates/supervisor/CLAUDE.md)

supplement/prompts/supervisor$_{c}laude.txt$

B.4 Domain tier: the supervisor's role (templates/supervisor/SUPERVISOR.md)

supplement/prompts/supervisor$_{r}ole.txt$

B.5 Task tier: the executor's instructions (templates/executor/CLAUDE.md)

supplement/prompts/executor$_{c}laude.txt$

B.6 MLE-bench launch brief: executor

The common executor brief of the MLE-bench runs (experiment/briefs/executor_common.md); each cell's brief fills in the competition, its supervisor's name, the data path and

the metric. The three-hour extension and the 24-hour stage used expanded versions of these briefs with budget, timing and snapshot sections. supplement/prompts/mlebench$_{e}xecutor_{b}rief.txt$

B.7 MLE-BENCH LAUNCH BRIEF: SUPERVISOR

The common supervisor brief (`experiment/briefs/supervisor_common.md`); each cell's brief fills in the competition and the executor's name. supplement/prompts/mlebench$_{s}upervisor_{b}rief.txt$

C CORPUS EXAMPLES

This appendix follows one paper through the corpus: the paper whose note carried the rebuttal in the consultation of Section 3.1, *Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution* (Kumar et al., 2022). The three items are reproduced verbatim, with the typography converted to ASCII.

C.1 WHAT A CONSULTATION SEES FIRST: SEARCH RESULTS

The search tool returns about 250 bytes per paper: the title; the score and tier of Section 4.4, the year, the citation count, the venue and the paper's role; its topic and subtopic; the path of its note; and the note's one-line summary. A consultation opens a note only when this line does not answer the question, and then reads one section of it. supplement/corpus/lookup$_{e}xample.txt$

C.2 A NOTE IN FULL

Every note has the same front matter (title, year, venue, arXiv id, topic, subtopic, role, keywords) and the same seven sections in the same order: Problem, Key idea, Method details, Results, Limitations and criticism, Practical takeaways, Related work. Quantitative content comes only from the authors' abstract; the synthesis sections are the note-writer's reading of it (Section 4.1). The note carries no citation count, score or tier. supplement/corpus/note$_{l}p_{f}t.txt$

C.3 THE INDEX RECORD BEHIND THE NUMBERS

Citation counts, the per-year histogram, the score and the tier live in one index per topic (`pipeline/enriched/<topic>.json`), never in a note; every published number is regenerated from it. The title and the arXiv id come from arXiv, the citation record from the Semantic Scholar Graph API (Table 1). The record of this paper: supplement/corpus/index$_{r}ecord_{l}p_{f}t.txt$

D THE CONSULTATION TRACE IN FULL

Section 3.1 summarizes this thread; the record is given here in full. The labels appear in the syntax used at the time (`[DB]` with the note's path, `[DB-ungrounded]`).

The domain-supervisor session overseeing autonomous pose-estimation experiments in the same project (the domain tier of Section 3) was the consultation protocol's first live user. Its questions carried domain coordinates (cell configurations, folds), but the questions themselves were at the level of general deep learning (fine-tuning bimodality, representation rank, statistical power): the abstraction the tier interface is meant to produce. We summarize one thread. This is a deployment record, not a controlled evaluation; it is offered as an illustration of the asymmetries of Section 3 at work, not as a quantitative claim. For scale: at the time of this thread the deployment had served two domain supervisors across five consultation threads; the threads had produced seven candidate reports; all seven passed verification (arXiv id, title and authors checked against the claim) and were registered, so the gate had corrected metadata, never rejected a candidate outright, and its discriminative strength was untested. On the advisor side, the four single-session threads among these cost 48k–70k tokens each, end to end (the thread agent's total, its own reasoning included), which is what the tiered consumption path of Section 4.2 amounts to in practice at this corpus size.

The supervisor asked for a corpus-grounded reading of why a ViT with a single temporal token shows bimodal fine-tuning outcomes $\{0.73 \mid 0.41\}$. The advisor first marked the literal question (a mechanism by which token-structure impoverishment creates a bad basin) as `[DB-ungrounded]`, then laid out five grounded lines with `[DB]` and note paths (that fine-tuning can distort pretrained features (Kumar et al., 2022), that ViT training instability can hide behind plausible-looking results (Chen et al., 2021), effective-rank diagnostics on checkpoints (Garrido et al., 2023), among others), and attached a cost-ordered list of discriminating experiments. A question that arrived carrying one hypothesis left carrying five discriminable ones; that is the shape of advice the objective asymmetry produces.

In the next round the context asymmetry did its work. The supervisor ran the free diagnostic (a training-loss comparison), found the collapsed runs' training-loss floor about $2.2\times$ higher, and reported this as consistent with the feature-distortion hypothesis it favored. The advisor rebutted from note content: that paper's predicted signature is training loss *preserved* with damage appearing only out of distribution, so an elevated loss floor weakens the hypothesis and promotes the optimization-failure family. The supervisor re-ranked its hypotheses before spending GPU. In the same round the advisor pointed out that the planned $n{=}1$ versus $n{=}1$ comparison repeated the very inference error just corrected (one-sided Fisher's exact test $p{=}0.5$; with five seeds per arm, $p{=}1/252 \approx 0.004$), and the supervisor pre-registered a three-arm $\times$ five-seed design with fixed decision rules.

The corrections ran in both directions, and the advisor was itself a target. The supervisor's learning-rate logs showed that a resume-schedule concern the advisor had raised did not apply to this setup (constant learning rate throughout). More telling: the failure-cause sketch the advisor had offered as `[DB-ungrounded]` while the note did not yet exist was partly a memory error, and once the note was written the advisor corrected itself; the paper's actual list of causes was different. Grounding labels do not hide the advisor's errors; they mark in advance where errors can live, and that marking was the path of this self-correction.

The consultation also changed the corpus. Two gaps exposed while answering were reported as candidates (the canonical study of fine-tuning instability that splits across random seeds (Zhang et al., 2021), and the work establishing attention entropy collapse as a measurable training instability (Zhai et al., 2023)), verified by the owner against arXiv and registered; from the round after their notes were written, claims previously labeled `[DB-ungrounded]` became `[DB]`. One incidental observation: the former is an ICLR paper with 509 citations that lands in the lowest tier because its momentum has decayed: a live instance of the design distinction (Section 4.4) that the score is a reading-priority weight, not a quality verdict.

The thread was still live as this draft was written, and its later rounds showed the same structure holding under repetition. Pre-registered decision rules were frozen before outcomes were observed, and when the same $n{=}1$ inference error recurred a third time, it was the rules, not a person, that caught it. The symmetry of correction held as well: the advisor downgraded two of its own quantitative claims (a gradient-concentration factor that stopped holding once the true layout was known, and the evidential weight of a paired design whose premise a determinism check destroyed) before the data arrived, and had them withdrawn from the asker's records. And when one diagnostic missed both of its registered predictions, the result read as refutation rather than confirmation precisely because the predictions had been fixed in advance. This is how the disciplines of Section 3 live in practice: separation, labels, and pre-registration do not guarantee right answers; they make it hard for wrong ones to survive posing as right.