Towards Autonomous and Auditable Medical Imaging Model Development

Shengyuan Liu[1,*], **Jia-Xuan Jiang**[1,*], **Boyun Zheng**[1], **Cheng Wang**[1], **Zipei Wang**[2], **Wentao Pan**[1], **Hongtao Wu**[1], **Houwen Peng**[5], **Yu Gu**[3], **Lichao Sun**[4,†], **Yixuan Yuan**[1,†]

[1]The Chinese University of Hong Kong, [2]Institute of Automation, Chinese Academy of Sciences, [3]Microsoft Research, [4]Lehigh University, [5]Independent Researcher

[*]Equal contribution, [†]Corresponding author

Large anguage model (LLM) agents are beginning to automate machine learning engineering (MLE) by coupling planning, code execution, debugging, and empirical feedback. Translating this capability to medical imaging remains difficult because each task imposes modality-specific experimentation and strict requirements for validation protocols and prediction artifacts. Here we introduce AMID, an autonomous multi-agent framework for medical imaging model development. AMID first proposes Data-Conditioned Method Planning, which refines coarse task-level search spaces into executable, parallelizable method lanes grounded in task-specific data analysis and runnable medical-imaging resources. It then develops Verification-Guided Two-Stage Optimization, moving from broad early exploration of diverse method lanes to selective exploitation of promising candidates while enforcing strict verification of validation protocols, metric computation, and prediction artifacts throughout the optimization. Across 20 medical imaging challenge tasks spanning diverse modalities and prediction types, AMID outperformed evaluated general-purpose MLE systems and, on several tasks, approached or matched strong human-designed challenge solutions. These results suggest that AMID can turn task-specific medical imaging model development from bespoke manual engineering into an agentic workflow for producing high-performing and auditable model artifacts across heterogeneous tasks.

Note: This is an ongoing preliminary technical report.

Project page: https://github.com/CUHK-AIM-Group/AMID

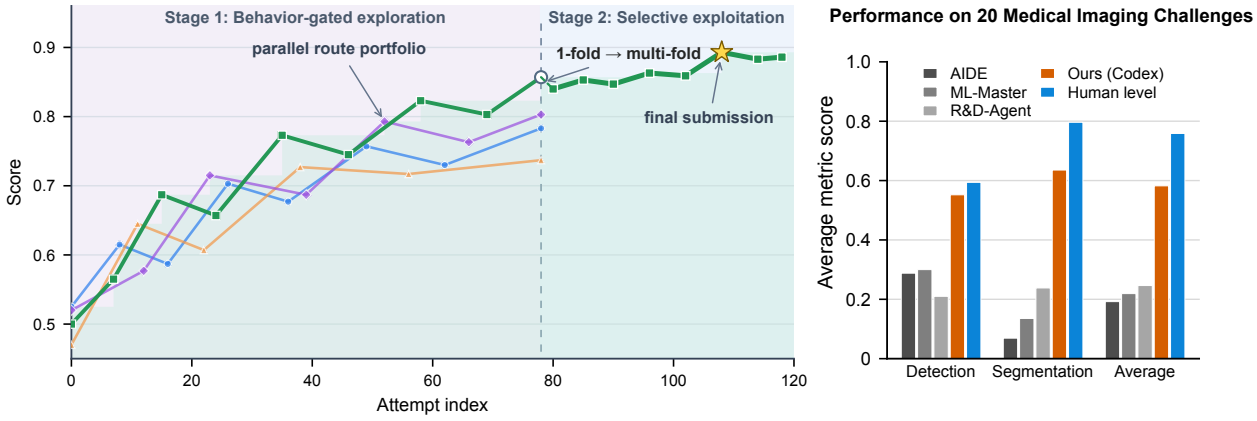

Figure 1 Overview of AMID's verification-guided optimization and benchmark performance. The two-stage optimization moves from broad exploration of parallel method lanes to selective exploitation of promoted candidates, while reviewer checks verify validation protocols, metric computation, and prediction artifacts throughout the optimization; shaded regions mark the two stages, and the star marks the final selected submission. The category-level bar chart reports average metric scores over 20 ReX-MLE medical imaging challenges.

1 Introduction

Large language model (LLM) agents are emerging as a promising paradigm for automating machine learning engineering (MLE), shifting model development from one-shot code generation toward iterative, feedback-driven experimentation (Aygün et al., 2026; Chan et al., 2024; Jimenez et al., 2024). In this paradigm, agents iteratively transform methodological ideas into executable experiments, using failures, validation signals, and accumulated evidence to guide subsequent refinements. Existing MLE agents (Jiang et al., 2025; Liu et al., 2025; Yang et al., 2025b; Du et al., 2025, 2026) have made this loop more structured and feedback-aware, coordinating exploration, execution, and evaluation while separating high-level methodological reasoning from code-level implementation and repair. Autonomous research agents extend this perspective beyond model development, showing that empirical loops must ultimately support scientific claims rather than merely produce improved results (Lu et al., 2024; Yamada et al., 2025; Weng et al., 2025; Yang et al., 2026; Li et al., 2026; Liu et al., 2026a). Together, these developments suggest that agentic MLE can substantially reduce the engineering burden of model development, but only when its search process is executable and auditable.

Translating agentic MLE to medical imaging is not simply a matter of applying a generic experimentation loop to clinical datasets (Huang et al., 2026a; Ferber et al., 2026). Medical imaging model development couples machine learning decisions with clinical and data-specific constraints, where the choice of preprocessing, model family, validation protocol, metric, and prediction artifact is shaped by the imaging modality, clinical objective, annotation process, and patient-level data organization (Litjens et al., 2017; Ronneberger et al., 2015; Isensee et al., 2021). As a result, an apparently standard prediction task often requires domain-conditioned engineering rather than generic pipeline search. Biomedical foundation models and generalist medical AI systems, including Med-PaLM M (Tu et al., 2023), BiomedGPT (Zhang et al., 2024), BioMedParse (Zhao et al., 2025), MedVersa (Zhou et al., 2026), and MedSAM (Ma et al., 2024a), provide reusable representations, segmentation priors, and multimodal reasoning capabilities, but they remain building blocks within a broader development workflow. They do not by themselves determine how a particular dataset should be preprocessed, modeled, evaluated, and adapted to its clinical objective (DeGrave et al., 2021; Geirhos et al., 2020; Gichoya et al., 2022; Yang et al., 2024; Ong Ly et al., 2024). Thus, medical imaging requires autonomous MLE systems that can reason from task-specific data and domain constraints, rather than merely reuse general-purpose modeling routines.

This requirement exposes two limitations of generic autonomous MLE systems. First, the search space constructed by generic agents is often too coarse-grained for heterogeneous medical-imaging tasks. A task may be recognized as classification, segmentation, detection, or restoration, but the appropriate method family can depend on modality, anatomy, acquisition protocol, annotation semantics, clinical target, and submission constraints. Without explicitly converting these medical-imaging attributes into executable and parallelizable method lanes, an agent may spend compute on pipelines that are technically executable but medically inappropriate, poorly matched to the data, or unable to produce valid final artifacts. Second, experimental feedback in medical imaging is often expensive, delayed, and potentially unreliable. Unlike many CPU-oriented or lightweight benchmark tasks where feedback can be obtained quickly, medical-imaging experiments may require high GPU budgets, long training runs, memory-aware inference, and strict patient-level or site-level splits. Under such conditions, agents may be biased toward low-cost shortcut experiments, incomplete training, weak validation protocols, or invalid artifacts. A single validation score can therefore be unstable or misleading under small cohorts, patient-level dependence, scanner or site effects, leakage, metric mismatch, fold mistakes, or invalid prediction artifacts, motivating a strict verification scheme that prevents untrusted evidence from driving the search process.

To address these challenges, we introduce **AMID**, an **A**utonomous multi-agent framework for **M**edical **I**maging model **D**evelopment. Given a medical-imaging task definition and its associated data, AMID aims to turn task-specific model development from labor-intensive manual engineering into an agentic workflow that produces high-performing, validated, and auditable model artifacts, including executable code, validation evidence, and submission-ready prediction artifacts. The framework is designed around a central premise: reliable medical MLE requires both explicit construction of the candidate method space and rigorous verification of the evidence used to select among candidates. First, AMID performs Data-Conditioned Method Planning (DCMP), which refines coarse task-level search spaces into executable, parallelizable method lanes grounded in task-specific data analysis and runnable medical-imaging resources. Rather than searching over generic pipelines, DCMP analyzes the modality, label structure, evaluation target, and submission requirements of each task, then grounds candidate methods in relevant code, literature, prior templates, and foundation models, organizing them into method lanes with clear assumptions, implementation plans,

and validation requirements. Second, AMID performs Verification-Guided Two-Stage Optimization, a hierarchical optimization procedure that moves from broad early exploration of diverse method lanes to selective exploitation of promising candidates. Throughout both stages, a reviewer process enforces strict verification of validation protocols, metric computation, and prediction or submission artifacts before any attempt can drive promotion or final selection. We conduct a comprehensive evaluation of AMID on 20 medical-imaging challenge tasks spanning diverse modalities and prediction types, demonstrating that AMID outperforms evaluated general-purpose MLE systems and, on several tasks, approaches or matches strong human-designed challenge solutions.

Our contributions can be summarized as:

- To our knowledge, AMID is the first autonomous multi-agent MLE framework designed for general medical imaging model development, producing auditable model packages with executable code, validation evidence, and submission-ready prediction artifacts across heterogeneous tasks and modalities.
- We propose Data-Conditioned Method Planning (DCMP), which refines coarse task-level search spaces into executable and parallelizable method lanes grounded in task-specific data analysis, relevant medical-imaging code, literature, prior templates, and foundation models.
- We introduce Verification-Guided Two-Stage Optimization, which moves from broad early exploration of diverse method lanes to selective exploitation of promising candidates while enforcing reviewer checks on validation protocols, metric computation, and prediction artifacts throughout the optimization process.

2 System Overview

As shown in Figure 2, AMID turns a medical-imaging task into an auditable model package through a data-conditioned and verification-controlled agent workflow. The system first establishes the task contract and profiles the input data, then converts the resulting evidence into executable method lanes, coordinates agents through a two-stage explore-and-exploit optimization, and accepts a final model or submission only when the supporting artifacts satisfy the task contract. This section follows the same system view: we first formalize the model-development problem, then describe how AMID plans method lanes, optimizes them through Verification-Guided Two-Stage Optimization, and preserves evidence across agent runtimes.

2.1 Task and Artifact Contract

AMID addresses a practical medical imaging model-development problem. The input is intentionally minimal: a medical-imaging dataset and a task definition that specifies the target output, evaluation metric, and, when applicable, the submission protocol. The dataset may contain 2D images, 3D volumes, pathology tiles, paired enhancement data, detection annotations, segmentation masks, class labels, graph labels, or image-quality scores. The task definition may come from a clinical modeling request or from a challenge description. The expected output is not a single text answer or a suggested architecture, but a complete model package: executable training and inference code, model weights or checkpoints, prediction files, validation scores, final submission artifacts when required, and an audit trail showing that the result was produced under the correct data, metric, split, and submission contract.

This framing treats autonomous medical imaging AI development as accountable engineering. A high local score is useful only if the score is comparable, the metric direction is correct, the split is legal, the prediction format matches the evaluator, and the model lineage can be inspected. AMID therefore searches for model-development procedures that are both high-performing and verifiable. The system is designed to start from basic task information and end with an auditable candidate model, while making intermediate choices inspectable to human reviewers.

2.2 Workflow Architecture

AMID organizes autonomous medical-imaging model building as a staged multi-agent run. The system first materializes a workspace from the task files and records the modeling target, evaluation metric, prediction schema, and required output contract. It then profiles the public data view and writes an evidence-backed summary of modality, dimensionality, supervision type, sample organization, geometry, label structure, and risks such as leakage, imbalance, sparse masks, or invalid output formats. Data-Conditioned Method Planning uses this profile to build a method-lane portfolio, grouping implementation-backed routes into method families for parallel exploration. Verification-Guided

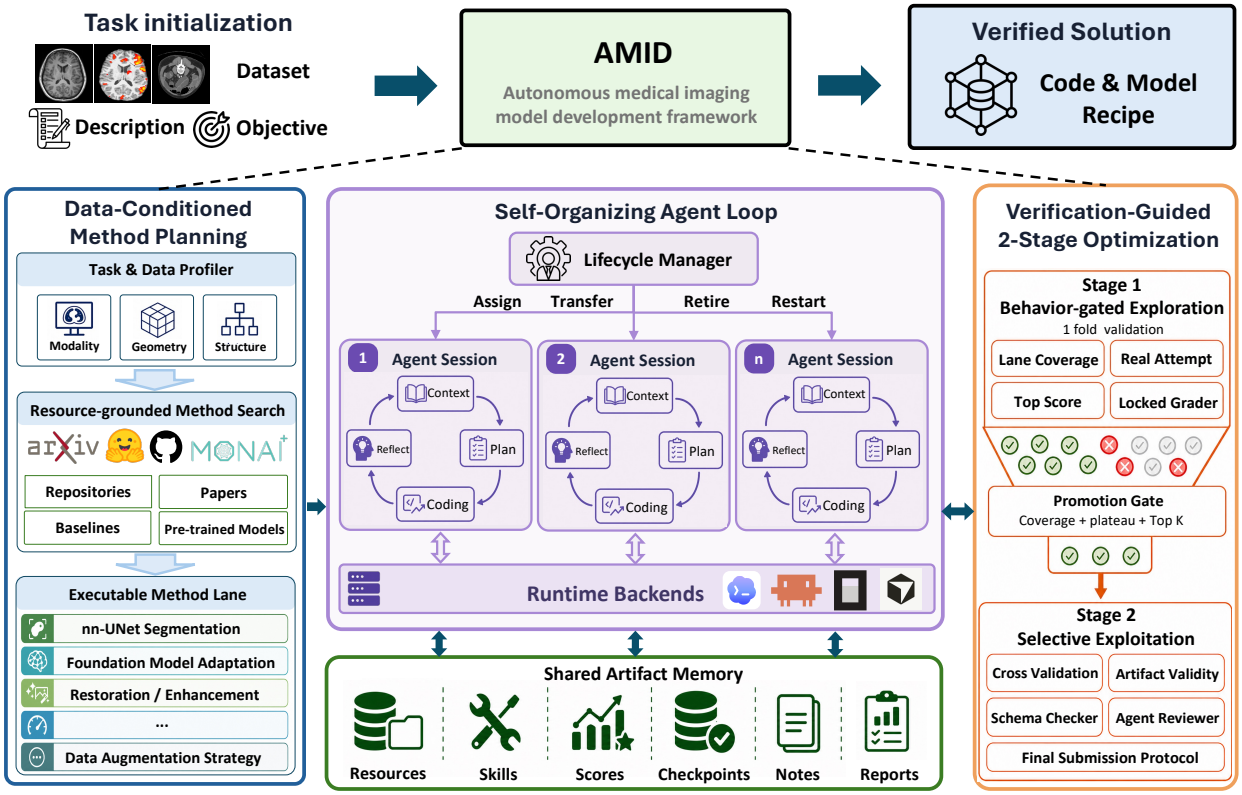

Figure 2 Overview of AMID. Starting from task initialization, AMID profiles the medical imaging data and converts task evidence into executable method lanes through Data-Conditioned Method Planning. A self-organizing agent loop then explores a diverse method portfolio, performs selective exploitation on promising candidates, and accepts a final model package only when reviewer-checked artifacts satisfy the task contract.

Two-Stage Optimization then turns this portfolio into an executable optimization process that moves from broad early exploration of diverse method lanes to selective exploitation of promising candidates. In both stages, reviewer checks verify the validation protocol, metric computation, and prediction or submission artifacts before an attempt can influence promotion, repair, or final selection. During exploration, workers cover different method lanes and produce comparable attempts with code, validation scores, logs, predictions, and submission artifacts. During selective exploitation, workers repair implementation issues, complete validation evidence, tune or ensemble selected variants when appropriate, and prepare the final deliverable from reviewer-accepted candidates.

This model-building workflow supports several deployment modes. In the default open-ended mode, agents continue improving candidate models until the user stops the run or accepts a result. AMID also supports a fixed-time-budget mode for settings such as competitions, where the manager schedules exploration, promotion, and finalization under an explicit deadline. The final deliverable is configurable: a task may require a submission file, prediction files referenced by that submission, an executable model package such as a Dockerized predictor, or both. The same search policy can be executed by different coding-agent backends, including Codex, Claude Code, OpenCode, Cursor Agent, and Kiro, and mixed-backend runs can allocate different agents to different runtimes while preserving the same shared artifacts, validation protocol, and final-artifact checks.

2.3 Data-Conditioned Method Planning

Data-Conditioned Method Planning (DCMP) addresses a failure mode we observe in general-purpose autonomous model-building agents. Medical imaging tasks are rarely solved well by selecting a generic training recipe from the task statement alone. The same nominal problem type can require very different choices depending on modality, dimensionality, voxel spacing, annotation density, object scale, patient grouping, metric definition, and output format. Conversely, a broad literature or web search often returns impressive methods that are not executable in the current setting because the code is unavailable, the checkpoint is gated, the data interface is incompatible, or the method

assumes a different imaging geometry. DCMP is designed to make planning conditional on the actual data and on resources that can be used by workers. It converts the task contract, public data evidence, and executable resource inventory into a portfolio of method lanes for search.

Task and data profiling. The first stage builds an evidence-backed profile of the public task view. AMID inspects task descriptions, README files, sample submissions, metric definitions or grader-facing hints, visible training data, labels, metadata, and the required prediction artifacts. This phase does not train models or search for methods. Its purpose is to make the dataset legible for later planning and to prevent each worker from rediscovering the same constraints.

The profile records several groups of planning-relevant evidence. At the inventory level, it summarizes file types, directory layout, train/test organization, sample counts, identifier structure, pairing relationships, missing or duplicated cases, and visible strata such as organ, site, scanner, view, phase, stain, or acquisition protocol when these are available. At the image-representation level, it samples raw files to estimate shape, channel count, dtype, intensity range, spacing, orientation, affine consistency, anisotropy, and volume-size distribution. At the supervision level, it characterizes the target type and label statistics: class balance for classification, foreground ratio and connected components for segmentation, box counts and object-size ranges for detection, alignment and contrast shifts for paired restoration, and modality missingness or registration relationships for multimodal tasks. At the evaluation level, it records the primary metric, metric direction, fold or group constraints, patient-level leakage risks, submission columns, referenced prediction files, and format checks that must hold before a score is trusted.

These diagnostics become search predicates. For example, sparse 3D vessel masks suggest patch sampling, spacing normalization, topology-aware post-processing, and memory-aware inference; histopathology data suggest tiling, stain variation handling, and slide- or patient-level aggregation; paired ultrasound enhancement requires alignment audits before pixel losses are trusted. The output is therefore not only a descriptive report, but a set of constraints and opportunities that downstream method search can use.

Resource-grounded method search. The second stage searches for methods under those data conditions. AMID distinguishes between two resource sources. The first is a pre-seeded local resource library that can be provided before a run. In our implementation, this library includes common medical-imaging toolkits such as nnU-Net (Isensee et al., 2021) and MONAI (Cardoso et al., 2022). nnU-Net provides a strong starting point for segmentation tasks that require task-specific dataset conversion, fold management, preprocessing, training, and inference. MONAI provides reusable medical-imaging components for data loading, spatial and intensity transforms, losses, network architectures, pretrained bundles, and deployment patterns across segmentation, classification, detection, restoration, and other tasks. AMID treats these resources as optional implementation anchors rather than mandatory baselines: a resource is assigned to a lane only when the profiled modality, task family, data geometry, and output contract make it actionable.

The second source is a curated foundation-model and external-resource search. AMID can consult a registry of medical-imaging foundation models indexed by domain, modality, task family, keywords, access status, license, model card, repository, and downloadable files. The registry includes, for example, promptable segmentation models such as MedSAM (Ma et al., 2024a) and MedSAM2 (Ma et al., 2025), anatomy-oriented tools such as TotalSegmentator (Wasserthal et al., 2023) and VISTA3D (He et al., 2025), and domain-specific foundation models for settings such as pathology, chest radiography, CT, MRI, and retinal imaging. For each candidate, the planner checks whether it matches the profiled data and whether its usable artifacts can be verified or acquired, so workers receive resources that are relevant to the task rather than a generic list of popular models.

DCMP combines the data profile and resource evidence into a hybrid method graph. Nodes represent data conditions, method families, and implementation resources; edges explain why a method family is plausible under the observed task constraints. The final lane portfolio is broader than a single best guess. It may include a conservative nnU-Net or MONAI baseline, a foundation-model adaptation lane, a task-specific architecture lane, a preprocessing- or post-processing-heavy lane, an ensemble lane, and calibration-only lanes that verify data loading, metric computation, and submission mechanics. Primary and secondary lanes must be implementation-backed. Lower-confidence, mismatched, paper-only, or over-budget ideas are retained as fallback lanes, downranked, or excluded rather than silently removed.

Each method lane contains a modeling hypothesis, the data conditions that motivate it, required resources, implementation requirements, preprocessing assumptions, validation obligations, approximate training budget, expected artifacts, and likely failure modes. Worker agents therefore start search from executable routes rather than from

open-ended brainstorming. This is the role of DCMP: it does not decide the final model, but it shapes the search space so that subsequent exploration spends effort on methods that are both medically plausible for the observed data and operationally runnable in the current workspace.

2.4 Verification-Guided Two-Stage Optimization

Medical-imaging model building often admits multiple qualitatively different solution families, and different families can dominate under different data conditions. A task may be approached through a robust nnU-Net-style pipeline, a foundation-model adaptation, a task-specific architecture, or a preprocessing/post-processing-heavy route. This diversity makes early commitment to a single direction risky. It also requires strict verification beyond score-driven optimization alone: a high early score may come from a narrow split, leakage, a metric or formatting mismatch, or an implementation that cannot produce the required final artifact. AMID therefore uses Verification-Guided Two-Stage Optimization over the method-lane portfolio produced by DCMP. The optimization units are lanes grouped into method families, where each lane specifies a concrete modeling hypothesis, resource set, and validation obligation. The process moves from broad early exploration of diverse method lanes to selective exploitation of promising candidates. In both stages, a reviewer checks whether each submitted attempt satisfies the active validation protocol, computes the metric correctly, and produces valid prediction or submission artifacts before the evidence can drive promotion or final selection.

Stage 1: Behavior-gated Exploration. The first stage gives AMID a portfolio-level view of the DCMP search space. Worker agents are distributed across method families and submit attempts tied to specific method lanes. Each attempt records the committed code, predictions or submission artifacts, validation score, metadata, and lane attribution. AMID uses these records to estimate which lanes are feasible, which method families show empirical promise, and which failure modes recur across implementations. The goal is not to exhaust every possible variant, but to obtain enough comparable behavior from the major lanes before committing the search to a small set of candidates.

This exploration is behavior-gated by a reviewer that is independent of the worker agents. Workers can propose an implementation plan for a lane and submit code, but they cannot certify their own evidence. Each submission is first recorded as a pending attempt with its committed code, worker identity, lane metadata, shared-state checkpoint, declared budget class, and produced artifacts. The reviewer then evaluates the frozen commit outside the worker session, checking whether the implementation plan was actually carried out under the active protocol and whether the resulting evidence is admissible. These checks cover the validation split, metric direction and computation, complete predictions or submission files, non-duplicate payloads, required logs or checkpoints, and traceability from code to artifacts. Incomplete runs, malformed predictions, duplicate payloads, tune-only checks, and metric-incompatible submissions remain useful for diagnosis, but they are not counted as valid lane coverage. Only attempts that pass this independent review become real, attributable, and comparable evidence. Once required lanes have enough reviewer-accepted attempts and show plateau behavior or a documented waiver, the manager ranks lanes by their best valid evidence and promotes the strongest candidates, optionally preserving a close second lane as a challenger.

Stage 2: Selective Exploitation. The second stage concentrates computation on the promoted candidates. Workers no longer perform broad method discovery; they start from accepted or promoted anchors and convert them into complete, auditable model packages. Optimizer workers improve the leading recipe through model, training, inference, test-time augmentation, post-processing, or ensemble changes. Challenger workers preserve limited capacity for a distinct promoted lane when the first-stage evidence does not clearly identify a single winner. Repair workers handle candidates that have useful raw scores but fail protocol or artifact checks, while finalizer workers prepare the final prediction, submission, or model package from an accepted anchor.

The same reviewer gate remains active during selective exploitation. A stage-2 candidate must satisfy the active validation protocol and produce evidence that can be inspected after the run. For medical-imaging tasks, this includes correct metric direction, legal data access, locked-fold usage when applicable, complete fold outputs, correct mean-score computation, fold provenance, fold-invariant pipelines, valid prediction schema, complete required files, and traceability from the final package back to an accepted attempt. Candidates that fail these checks may still inform repair or further optimization, but they are not admitted to the accepted candidate set or selected as the final model. The final package is therefore chosen from attempts that are both high-performing and reviewer-verified.

2.5 Self-Organizing Agent Loop

AMID adopts a self-organizing execution substrate and specializes it for medical-imaging model construction. The goal is to let multiple coding agents search asynchronously without hiding evidence inside private chat histories. The substrate is backend-agnostic: agents implemented with Codex, Claude Code, OpenCode, Cursor Agent, Kiro, or mixed backends can participate in the same run. Each worker runs in an isolated git worktree, but all workers operate against the same task contract, method lanes, attempts, notes, skills, resources, and audit artifacts. Unlike a fully peer-to-peer agent society, however, AMID places a central lifecycle manager around this substrate. Agents remain autonomous in implementation and debugging, but their creation, context, interruption, reassignment, retirement, and finalization are controlled by the manager.

Lifecycle manager. The lifecycle manager serves as the operational control plane of the agent loop. It maintains an explicit state for each worker, covering initialization, code editing, evaluation waiting, feedback reflection, reassignment, retirement, and artifact preparation. At launch time, the manager creates the worker workspace, attaches the shared state view, injects the task contract and current lane context, and starts the selected coding-agent backend. During execution, it monitors process health, recent attempts, idle time, context pressure, and signs of diminishing progress. When intervention is needed, the manager can interrupt and resume an existing session with a targeted prompt, rather than replacing it with a fresh conversation.

This lifecycle policy gives the system a recoverable outer structure. Worker sessions may fail, stall, exhaust their current direction, or approach context limits; in each case, the manager applies an explicit transition rather than leaving the population unmanaged. It can restart a session, backfill capacity, request documentation of a useful procedure, or shift a candidate-producing worker toward artifact completion. Thus, parallel agents are not merely launched and left to run independently. They are maintained as managed computational processes whose states, transitions, and responsibilities remain visible to the system.

Manager-mediated self-organization. Building on this lifecycle machinery, AMID realizes self-organization as an evidence-driven allocation of research effort rather than as coordination among agents with fixed identities. The unit of allocation is a temporary exploration focus assigned to a worker session. Such a focus denotes a direction of investigation rather than a durable role: it may correspond to a conservative baseline, a foundation-model adaptation, a data-processing route, a post-processing route, or a finalization path. Because foci are temporary and state-dependent, the worker population can be continuously reconfigured as the run unfolds. The resulting organization is therefore not a static ensemble of specialists, but a mutable distribution of effort over competing directions of progress.

The worker loop in Figure 2 provides the local evidence used for this reconfiguration. Each worker turns shared context into a bounded implementation plan, tests that plan in its own worktree, and externalizes the outcome through a public reflection. The manager aggregates these public traces to decide how collective attention should be redistributed: productive directions can be reinforced, redundant efforts merged, saturated branches deprioritized, and exploratory threads preserved when diversity remains useful. This yields a two-level organization of search: workers exercise local autonomy over implementation, debugging, and evaluation, while the manager performs global, evidence-driven allocation of research attention.

Shared memory. The shared memory is a file-system state rather than an in-context transcript. AMID stores attempt records, experiment notes, method-lane files, resource manifests, validation state, reviewer reports, final-artifact records, manager state, and reusable skills in a shared workspace visible to all agents. Attempt records bind a score to a commit hash, agent identity, validation protocol, method-lane attribution, artifact paths, and grader feedback. Notes record observations such as preprocessing failures, fold-specific behavior, resource blockers, or promising next experiments. Because this state is durable and shared, later workers can inspect what has already been tried, reuse successful code paths, avoid known failures, and audit why a candidate was promoted or rejected.

Skills are treated as a structured part of this memory. AMID seeds each run with skills for organizing files, using reference repositories, reading papers for reproduction, creating new skills, and adapting foundation models. The foundation-model skill, for example, does not select a model by itself; it is invoked after DCMP or a worker chooses a specific model, checkpoint, or model-zoo bundle. It then verifies the local resource, reads the model card or configuration, maps the model interface to the current task, preserves the task-native data, fold, and evaluation pipeline, records provenance, and runs smoke tests before expensive training. Agents can also create or update skills

when a repeated workflow becomes useful. In this way, shared memory stores not only results, but also procedures that allow successful behavior to diffuse across workers during the same run.

Heartbeat interventions. AMID uses heartbeat prompts to keep long-running agents aligned with the search objectives. The manager monitors the attempt stream, per-agent progress, global evaluation count, stalled workers, and plateau behavior. When a heartbeat triggers, the manager interrupts and resumes the agent with a targeted prompt rather than resetting the session. Reflection heartbeats ask agents to convert recent results into experiment notes with concrete causes and next actions. Consolidation heartbeats ask agents to synthesize shared notes, identify contradictions, summarize team state, and expose uncovered or over-exploited directions. Plateau heartbeats force a pivot decision when an agent has stopped improving, requiring a public focus note that names the current focus, evidence, remaining budget, and abandon condition. In AMID, these prompts are tied to medical-imaging obligations: agents are reminded to preserve task-native data handling, avoid local tuning traps, inspect reference-backed directions before abandoning them, and maintain final-artifact evidence.

3 Experiments

3.1 Experimental Setup

To evaluate AMID's ability to develop competitive medical imaging models, we use ReX-MLE (Kenia et al., 2025), a challenge benchmark with clear task definitions, standardized metrics, and competitive reference points. ReX-MLE contains 20 medical imaging tasks drawn from MICCAI-style and related evaluation settings. The suite covers segmentation, detection, classification, image-quality assessment, and enhancement across panoramic X-ray, CT, MRI, CTA, MRA, histopathology, ultrasound, and microscopy data. Each task provides public data, a task description, a scoring metric, and a required prediction or submission format, allowing us to assess whether an agent can build a valid model pipeline from raw files to final artifacts. Unless otherwise specified, the main experiments use the Codex backend with GPT-5.5 as the underlying coding agent. For each challenge, AMID is given a 24-hour wall-clock budget and one NVIDIA RTX A6000 GPU with 48 GB of memory.

3.2 Main Results and Comparative Analysis

Table 1 gives the main comparison across the full 20-task ReX-MLE suite. We compare AMID with the three autonomous MLE systems reported in ReX-MLE: AIDE (Jiang et al., 2025), ML-Master (Liu et al., 2025), and R&D-Agent (Yang et al., 2025b). To keep the comparison focused on system design rather than metric reinterpretation, the baseline columns use the accepted benchmark scores reported by ReX-MLE under its original challenge contracts, while AMID is evaluated with the same accepted-artifact criterion. The comparison also uses GPT-family backends on both sides: the ReX-MLE baselines use GPT as their primary backend, and our main AMID run uses Codex with GPT-5.5. Thus, each entry reflects an end-to-end result, not a local validation number that might be invalidated by prediction format, metric direction, split discipline, or final submission packaging. The Human column is included only as a reference scale. These values are the original competition reference scores obtained on the true challenge test sets, which are not publicly released. ReX-MLE therefore constructs its evaluation by re-splitting the originally public training data; agents in this setting train with less data than teams had in the original competitions. Human-normalized comparisons should consequently be read as approximate performance references rather than strictly matched leaderboard comparisons.

Across the 20 challenges, AMID produces valid accepted results for every task and improves over the strongest listed baseline on 19 tasks, with the remaining task tied on TopCoW-CTA-Cls. The gains are most pronounced on segmentation and detection tasks, where completing the medical-imaging pipeline is often the main bottleneck for general-purpose agents. In segmentation, AMID raises SEG.A Dice by +0.89 and ISLES'22 Dice by +0.67 over the best baseline, while also converting previously failed or near-zero PUMA tissue-segmentation runs into valid Dice scores above 0.5. In detection, AMID improves DENTEX AP by +0.40 and the PUMA-T1 nuclei detection F1 score by +0.46. Relative to the human reference scale in Figure 3, the strongest AMID results reach or exceed the reference on DENTEX, NeurIPS-CellSeg, PUMA-T2-Det, and LDCT-IQA, and come close on SEG.A. Performance remains weaker on graph classification, vessel-analysis tasks that depend on small anatomical structures, and ultrasound enhancement. These gaps show that AMID is broadly stronger than existing autonomous MLE baselines, but expert-level performance remains task-dependent rather than automatic.

Table 1 Main ReX-MLE benchmark comparison across 20 medical imaging challenges. Challenge names and citations follow the ReX-MLE benchmark overview. All entries report accepted primary metric values in the original challenge units; Ours denotes AMID with the Codex+GPT-5.5 backend, Δ reports the absolute improvement over the strongest listed agent baseline, and Human gives original-competition reference performance rather than a matched ReX-MLE split score.

Challenge	Metric (↑)	AIDE (Jiang et al., 2025)	ML-Master (Liu et al., 2025)	R&D-Agent (Yang et al., 2025b)	Ours	Δ	Human
Segmentation Tasks							
ISLES'22 (Hernandez Petzsche et al., 2022)	Dice	0.04	0.00	0.02	**0.71**	+0.67	0.79
NeurIPS-CellSeg (Ma et al., 2024b)	F1	0.04	0.04	0.36	**0.90**	+0.54	0.88
PANTHER-T1 (Betancourt Tarifa et al., 2025)	Dice	0.33	0.13	0.16	**0.42**	+0.09	0.73
PANTHER-T2 (Betancourt Tarifa et al., 2025)	Dice	0.09	0.05	0.28	**0.31**	+0.03	0.53
PUMA-Track1-TissueSeg (Schuiveling et al., 2025)	Dice	**FAIL**	0.00	0.00	**0.56**	+0.56	0.78
PUMA-Track2-TissueSeg (Schuiveling et al., 2025)	Dice	0.00	0.00	0.00	**0.56**	+0.56	0.78
SEG.A (Jin et al., 2025)	Dice	0.02	0.02	0.00	**0.91**	+0.89	0.92
TopBrain-CTA-Seg (Yang et al., 2025a)	Mean Dice	0.03	0.26	0.08	**0.59**	+0.33	0.79
TopBrain-MRA-Seg (Yang et al., 2025a)	Mean Dice	0.01	0.26	0.50	**0.64**	+0.14	0.81
TopCoW-CTA-Seg (Yang et al., 2025a)	Mean Dice	0.09	0.25	0.49	**0.63**	+0.14	0.87
TopCoW-MRA-Seg (Yang et al., 2025a)	Mean Dice	0.11	0.48	0.73	**0.76**	+0.03	0.88
Detection Tasks							
DENTEX (Hamamci et al., 2023)	AP	0.09	0.08	0.09	**0.49**	+0.40	0.40
PUMA-Track1-NucleiDet (Schuiveling et al., 2025)	F1	0.02	0.08	0.06	**0.54**	+0.46	0.66
PUMA-Track2-NucleiDet (Schuiveling et al., 2025)	F1	**FAIL**	0.00	0.01	**0.28**	+0.27	0.27
TopCoW-CTA-Det (Yang et al., 2025a)	IoU	0.67	0.65	0.70	**0.71**	+0.01	0.79
TopCoW-MRA-Det (Yang et al., 2025a)	IoU	0.66	0.69	0.19	**0.74**	+0.05	0.85
Classification Tasks							
TopCoW-CTA-Cls (Yang et al., 2025a)	Accuracy	0.33	0.10	0.28	**0.33**	+0.00	0.73
TopCoW-MRA-Cls (Yang et al., 2025a)	Accuracy	0.33	0.33	0.09	**0.46**	+0.13	0.89
Image Quality & Enhancement Tasks							
LDCT-IQA (Lee et al., 2025)	Score	2.62	2.50	2.66	**2.74**	+0.08	2.74
USenhance (Guo et al., 2023)	LNCC	0.11	0.13	**FAIL**	**0.19**	+0.06	0.91

Table 2 Focused comparison on three representative ReX-MLE challenges. All metrics are higher-is-better. The first three rows report agent baselines under a unified GPT-5.5 model backend; the final two rows report AMID under Codex+GPT-5.5 and Claude Code runtime settings for the same challenge definitions.

Method	Model backend	DENTEX AP (↑)	PUMA-T1-Seg Dice (↑)	TopCoW-MRA-Cls Acc. (↑)
AIDE (Jiang et al., 2025)	GPT-5.5	0.08	0.39	**FAIL**
ML-Master (Liu et al., 2025)	GPT-5.5	0.06	0.50	0.39
R&D-Agent (Yang et al., 2025b)	GPT-5.5	0.08	0.52	0.09
Ours (Codex)	GPT-5.5	**0.49**	0.56	0.46
Ours (Claude Code)	Opus-4.8	0.25	**0.64**	**0.50**

Unified model-backend comparison. Table 2 complements the full sweep with a controlled view on three representative challenges. The first three rows hold the underlying model backend fixed at GPT-5.5 for AIDE, ML-Master, and R&D-Agent. Even under this stronger and unified model setting, the best baseline scores remain below the best AMID row on each representative task: 0.08 AP versus 0.49 AP on DENTEX, 0.52 Dice versus 0.64 Dice on PUMA-T1-Seg, and 0.39 accuracy versus 0.50 accuracy on TopCoW-MRA-Cls. This comparison rules out a weak-baseline explanation for the focused gaps and suggests that the higher-level search organization, evidence tracking, and final-artifact verification are important beyond the model backend alone.

Runtime-backend behavior within AMID. The final two rows of Table 2 show AMID under different model and runtime configurations. The Codex+GPT-5.5 row uses the same model backend and runtime family as the main full-suite run, while the Claude Code row probes the same AMID search layer under a different runtime and model

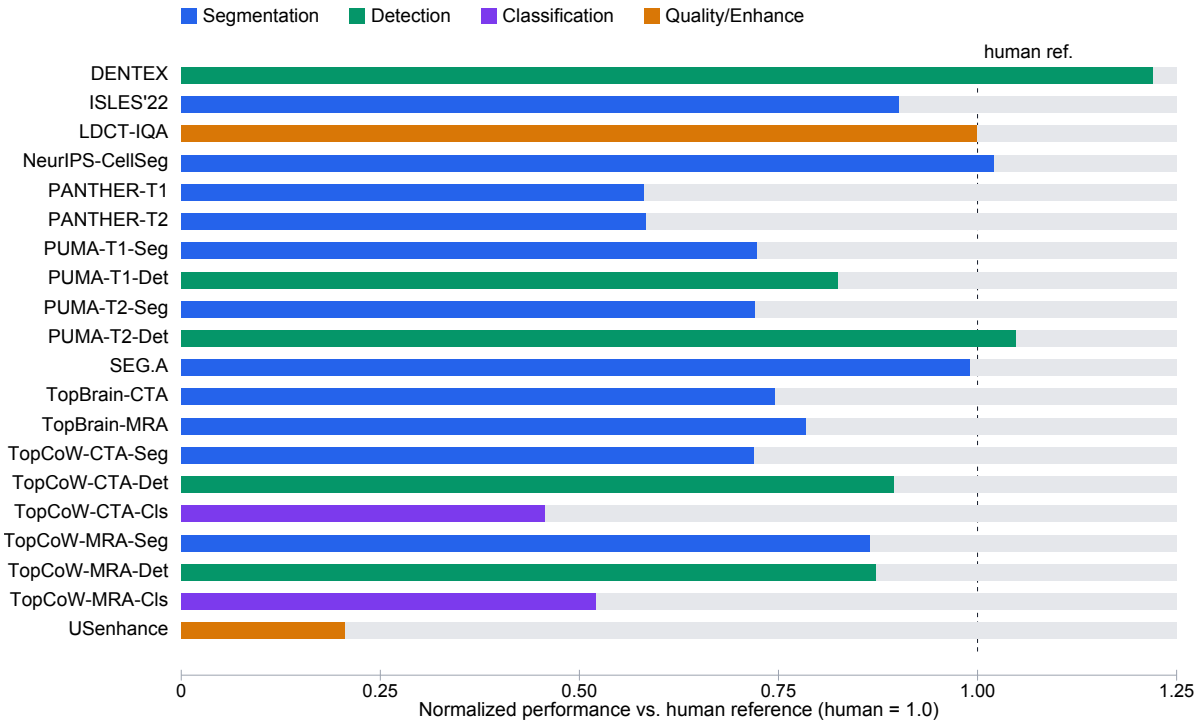

Figure 3 Challenge-wise comparison with human reference performance on ReX-MLE. Bars show accepted AMID performance normalized by each challenge's original-competition human reference; the dashed line marks reference parity and is used as an approximate scale rather than as a strictly matched leaderboard target.

backend. Codex+GPT-5.5 is strongest on DENTEX, reaching 0.49 AP, while Claude Code obtains the best focused scores on PUMA-T1-Seg and TopCoW-MRA-Cls, with 0.64 Dice and 0.50 accuracy. This mixed pattern indicates that runtime backends still affect individual implementation trajectories, debugging behavior, and task-specific method choices. However, the consistent advantage of the AMID rows over the corresponding baseline rows under the same challenge definitions supports the main claim: AMID's gains come from organizing and verifying the model-building process, not from a single favorable runtime alone.

3.3 Case Studies

Strong cases: search and task-specific adaptation. In this section, we analyze why AMID performs well in representative strong cases: broad search identifies task-matched methods, task-specific adaptation converts them into valid challenge solutions, and artifact verification keeps the final outputs aligned with the submission contract. DENTEX is a representative example. The final submission does not rely on a detector alone: it starts from a compact YOLOv8 (Varghese and Sambath, 2024) detector adapted to panoramic dental radiographs, then adds an anatomical calibration layer that maps unordered boxes to FDI quadrants and tooth numbers using dental-arch templates, conservative diagnosis fallback statistics, and learned alternate tooth-position rules. This hybrid design reaches 0.48825 mean AP, 0.64702 AP50, and 0.68248 AR, outperforming the autonomous-agent baselines in Table 1. The gain comes from matching the task structure: tooth enumeration is partly visual detection and partly anatomical ordering, so the selected solution combines a learned detector with structured domain priors instead of treating all labels as independent detector classes.

The PUMA pathology tasks show a second form of the same behavior. For tissue segmentation, AMID discovers pathology foundation encoders and uses them as the stable representation layer: PUMA Track 1 uses a locked five-fold ensemble over frozen UNI (Chen et al., 2024) patch-token features, where lightweight MLP heads combine local tissue tokens with RGB and coordinate cues; PUMA Track 2 fine-tunes the final Phikon-v2 (Filiot et al., 2024) block and blends full-image, fixed-crop, and overlapping-crop evidence. These choices convert tasks where prior autonomous agents failed or scored zero into accepted Dice scores of 0.56395 and 0.56214. For nuclei detection, the system chooses

a different decomposition. PUMA Track 1 first uses CellViT-256 (Hörst et al., 2024) to propose nuclei and token embeddings, then trains a deterministic gradient-boosting classifier with local and neighborhood context; PUMA Track 2 uses PathoSAM (Griebel et al., 2025) proposals and Midnight pathology embeddings with a small classical classifier ensemble. The resulting macro-F1 scores, 0.54359 for the three-class task and 0.28422 for the ten-class task, are especially informative because the Track 2 result ranks first in the official track despite a low absolute macro-F1 caused by rare cell classes. In both cases, broad method search matters: the selected pipelines are built around domain foundation models and task-specific post-processing, not around generic CNN training from scratch.

The broader sweep supports the same interpretation. On NeurIPS-CellSeg, AMID fine-tunes Cellpose-SAM (Pachitariu et al., 2025) and reaches a 0.893655 internal accepted score with official F1 values near 0.90 at IoU 0.5. On SEG.A and ISLES'22, it falls back to strong task-appropriate medical segmentation baselines, using nnU-Net-style (Isensee et al., 2021) planning, late-checkpoint selection, and conservative anatomical cleanup or recall-oriented mask unions. On TopCoW detection, it uses geometry priors, heatmap center prediction, segmentation-derived boxes, and lightweight direct detectors rather than forcing one generic detector across CTA and MRA. These traces illustrate why the two-stage optimization is useful: exploration supplies several plausible method lanes, while exploitation and artifact verification select the lane that both scores well and satisfies the exact submission contract.

Failure study. USenhance highlights a limitation of long-horizon image translation rather than a simple implementation error. The submitted five-fold pix2pix-style (Isola et al., 2017) U-Net ensemble is complete and produces valid outputs, but its official result remains low, with LNCC 0.18658, SSIM 0.38495, and PSNR 17.12416. Stronger CycleGAN-style (Zhu et al., 2017) adversarial candidates were difficult to promote because generator-discriminator losses were unstable, and early loss curves converged slowly. Under a fixed budget, these candidates therefore received weak short-horizon evidence and were rejected in favor of the more stable paired U-Net recipe, even though the challenge likely needs longer, carefully monitored training to balance anatomical fidelity, speckle realism, and local-normalized-correlation objectives.

The two TopCoW classification tasks expose a coupled segmentation-classification bottleneck. Although the official target is graph-edge classification, useful labels depend on an intermediate vessel segmentation and topology extraction for small structures such as PCom, ACom, and a third A2 segment. The CTA run collapses to a fixed topology prior, while the MRA run uses SegResNet vessel masks and contact rules but still reaches only 0.46296 anterior accuracy and 0.16667 posterior accuracy. This contrasts with the stronger pure TopCoW segmentation runs, suggesting that voxel-level vessel masks are not enough when the downstream objective depends on small branch contacts and graph labels. These tasks require segmentation, topology post-processing, and classification-oriented selection to improve together, making them harder than isolated segmentation or detection under the same budget.

We make the solutions obtained for all challenges publicly available on the project page.

4 Discussion

Medical model development as a search problem. AMID treats autonomous medical-imaging model development as a search problem over executable and verifiable solution candidates rather than as free-form idea generation or a single prompt-to-code task. This distinction is important because a useful candidate in medical imaging is more than a plausible architecture; it must match the modality, anatomy, annotation semantics, patient or case organization, metric direction, data scale, compute budget, and final artifact contract. Data-Conditioned Method Planning therefore turns the task and data profile into a portfolio of method lanes, where each lane carries a modeling hypothesis, implementation resources, feasibility assumptions, cost expectations, validation obligations, and likely failure modes. In this sense, AMID's "idea generation" step is deliberately closer to a domain-conditioned search system: it expands multiple medically plausible routes, downranks paper-only or incompatible resources, and makes the candidate space inspectable before expensive experimentation.

Verification as the control layer. The results also suggest that autonomous medical MLE should be evaluated based on verified evidence rather than on whether generated code happens to run. A runnable script can still use the wrong split, optimize the wrong metric, leak patient-level information, write malformed prediction files, duplicate a previous payload, or report a score that cannot be traced to a final submission. AMID addresses this failure mode by attaching acceptance criteria to each attempt: validation scores must be linked to committed code, logs, fold or

protocol records, metric computation, prediction artifacts, and lane attribution. In both stages, reviewer checks act as an internal validity layer, checking metric direction, legal data access, fold provenance, complete outputs, prediction schema, and traceability from accepted attempts to final packages. This verification layer is not only a final safety check after optimization. It changes the optimization itself, because candidates that cannot satisfy the protocol are used for diagnosis or repair but are not allowed to drive promotion or final selection.

From fine-grained prompting to agent-loop orchestration. AMID also reflects a shift in how long-horizon agents can be engineered. Earlier agent systems often relied on finely designed prompts for individual steps such as planning, coding, debugging, and summarization. Modern coding-agent runtimes already provide much of this execution harness: persistent workspaces, terminals, version control, tool use, and iterative repair. The remaining design problem is therefore less about hand-writing every low-level instruction and more about building the higher-level substrate around the harness. In AMID, this substrate consists of the task contract, shared file-system memory, method-lane records, attempt ledgers, heartbeat interventions, reusable skills, lifecycle scheduling, and final-artifact gates. These components let heterogeneous coding agents operate as workers inside the same auditable search process, while the system-level loop decides which lanes need coverage, which candidates merit exploitation, and which claims are supported by preserved evidence.

Scope beyond challenge submissions. The challenge benchmark setting is useful because it provides clear metrics, standardized artifacts, and competitive reference points, but the same design is not tied to challenge participation. The target deliverable can be a submission file, a trained model package, a Dockerized inference pipeline, a prediction service, or an evidence bundle for human review. Fixed-time-budget evaluation is one reproducible deployment mode; open-ended use can instead allow users to inspect evidence, stop a run, request additional exploration, or select among credible candidates. At the same time, challenge performance should be interpreted as system-level evidence rather than as clinical validation. It measures task-specific predictive performance under a defined evaluation protocol, but it does not establish clinical safety, prospective generalization, calibration under deployment shift, or suitability for patient care. AMID's artifact-first design is meant to make such later validation easier by preserving how a model was built, which resources it used, and where remaining risks may enter.

Limitations and next steps. The main limitation is cost. Long-horizon medical MLE consumes both GPU time and large numbers of coding-agent tokens, especially when the system explores multiple method lanes, runs stage-level reviewer checks, and preserves evidence across workers. For this reason, our current experiments do not exhaustively evaluate every challenge under all of the latest model-runtime combinations; the full-suite results use the primary backend, while newer or alternative backends are examined on focused representative tasks. Broader sweeps across model backends, compute budgets, method-lane ablations, verification ablations, and reviewer-style validity analyses remain necessary to separate the contribution of the search design from the strength of the underlying agent runtime. AMID also depends on the quality of the resource registry, the accuracy of data profiling, and the backend agent's ability to implement strong methods under budget. A key research direction is therefore cost-aware long-horizon autonomy: using cheaper early filters, stronger stopping rules, reusable skill distillation, cached resource audits, surrogate validation, and selective escalation to expensive models only when a candidate is worth deeper verification.

5 Related Work

Autonomous empirical software and MLE agents. Autonomous empirical software agents define progress through executable artifacts rather than through static text generation. Expert-level empirical software writing (Aygün et al., 2026), code-space exploration (Jiang et al., 2025), and AI-driven systems research (Cheng et al., 2025) all rely on an execution-feedback loop in which code, logs, tests, and validation scores become the evidence for subsequent decisions. SWE-bench (Jimenez et al., 2024) formalizes execution-feedback evaluation for repository-level software repair, whereas MLE-bench (Chan et al., 2024) and MLAgentBench (Huang et al., 2023) formalize autonomous model development as the production of training code, validation evidence, debugging traces, and final submissions. Recent benchmarks extend executable evaluation from general model building to specialized long-horizon AI engineering settings. PostTrainBench (Rank et al., 2026) focuses on autonomous LLM post-training under bounded compute, while MLS-Bench (Lyu et al., 2026) evaluates whether agents can propose, test, and validate ML improvements that generalize beyond a single tuned pipeline. Within benchmarked MLE settings, recent agent frameworks increasingly represent model development as structured search over runnable pipelines. ML-Master (Liu et al., 2025) uses MCTS-style search

to maintain competing solution branches, while AutoMLGen (Du et al., 2025) extends branch-level search with graph structure, intra-branch evolution, cross-branch reference, and multi-branch aggregation so that partial findings can propagate across the search graph. MLEvolve (Du et al., 2026) translates graph/tree search into specialized draft, improvement, debugging, evolution, fusion, result-parsing, and code-review agents, while Arbor (Jin et al., 2026) stores hypotheses, evidence, branches, and distilled insights in a persistent hypothesis tree. Recent studies (Zhang et al., 2026; Qu et al., 2026; Gao et al., 2026a) further treat general coding-agent runtimes, such as Claude Code or Codex, as substrates for higher-level autonomous systems, using runtime-backed repositories, shared persistent memory, heartbeat interventions, and self-organizing teams to support long-running experimentation.

Autonomous research systems. Autonomous research systems generalize empirical software loops into end-to-end scientific workflows encompassing hypothesis generation, literature grounding, experiment design and execution, result interpretation, manuscript drafting, and review. A growing body of research-agent benchmarks (Starace et al., 2025; Kim et al., 2025; Siegel et al., 2024; Hu et al., 2025; Ye et al., 2025; Huang et al., 2026b; Faroughy et al., 2026; Wang et al., 2026b) has operationalized this agenda through paper-grounded reproduction, replication, workflow reconstruction, and insight rediscovery tasks, revealing persistent limitations in faithful execution, long-horizon experiment management, and evidence-grounded reasoning that motivate more structured autonomous research systems. In parallel, early studies of autonomous scientific research and tool-augmented scientific reasoning (Boiko et al., 2023; Wang et al., 2024) showed that language-model agents can coordinate scientific actions beyond static question answering. Building on this premise, The AI Scientist (Lu et al., 2024) provided an early end-to-end formulation of idea-to-paper automation by mapping research ideas into experiments and manuscripts within constrained templates, while AI Scientist-v2 (Yamada et al., 2025) strengthened this formulation through agentic tree search and review-aware iteration. This line of work has since expanded the design space along several complementary dimensions: AI co-scientist (Gottweis et al., 2025) emphasizes multi-agent hypothesis generation, whereas CodeScientist (Jansen et al., 2025) grounds scientific discovery jointly in literature and executable code; related systems (Pu et al., 2025; Ifargan et al., 2025; Schmidgall et al., 2025; Tang et al., 2025; Weng et al., 2025) explore principle-guided exploration, evidence-aware manuscript generation, human-in-the-loop assistance, and autonomous research ideation. Another strand addresses the operational bottlenecks of idea-to-paper pipelines by equipping agents with more durable execution substrates: MARS (Chen et al., 2026) and AutoSOTA (Li et al., 2026) frame empirical AI research as repository-level optimization over papers, code, dependencies, compute budgets, reusable skills, and reproducible performance gains, while ScientistOne (Meng et al., 2026) elevates claim-level provenance to a first-class design constraint by tracking chains of evidence across literature, experiments, code, and writing. Complementing these efforts, AutoResearchClaw (Liu et al., 2026a) and ARIS (Yang et al., 2026) incorporate reusable skills directly into the research state, enabling agents to retrieve, adapt, and accumulate procedures for experiment repair, evidence checking, and manuscript construction rather than relying on one-off prompting or fixed pipelines. Taken together, these developments mark a shift from isolated demonstrations of scientific agency toward infrastructure-aware, provenance-grounded, and skill-accumulating autonomous research systems capable of sustaining iterative scientific workflows.

Medical autoresearch systems. Medical autoresearch systems can lower the barrier for clinicians to translate clinical or imaging questions into usable models, analyses, and manuscript-ready evidence, provided that they produce runnable development workflows and auditable artifacts rather than clinical answers alone. ReX-MLE (Kenia et al., 2025) makes this requirement explicit by evaluating agents on medical-imaging challenge tasks that require preprocessing, model development, metric handling, and valid submissions. AutoMedBench (Liu et al., 2026b) and HealthAgentBench (Liu et al., 2026c) further demonstrate that current agents still struggle with end-to-end clinical workflows that combine large search spaces with compositional reasoning. To solve these problems, medical AI scientist systems (Wu et al., 2026) and Camyla (Gao et al., 2026b) extend autonomous research into biomedical discovery and medical image segmentation, while related neuroimaging and pathology systems (Wang et al., 2026a; Han et al., 2026; Trost et al., 2026) emphasize domain-specific scientific assistance, analysis, or discovery workflows rather than general medical-imaging model construction. A recent study (Zhao et al., 2026) uses coding agents to translate natural-language clinical requests into runnable development loops and refine models under clinician-specified priorities. However, its demonstrations remain limited to a small set of constrained tasks and leave open how an agent should optimize across competing modeling strategies and how the selected result should be verified before delivery. AMID focuses on this optimization-and-verification problem through Verification-Guided Two-Stage Optimization and artifact auditing, so model packages are promoted, rejected, or accepted based on executable evidence rather than

clinician steering alone.

6 Conclusion

In this technical report, we presented AMID, an autonomous multi-agent framework for verifiable medical imaging AI development. AMID formulates model building as a data-conditioned and verification-controlled search process: it establishes the task contract, profiles the medical-imaging data, constructs executable method lanes, coordinates broad exploration and selective exploitation, and accepts final outputs only when reviewer-checked evidence and artifacts satisfy the required protocol. Across the ReX-MLE medical-imaging challenge suite, this design improves over existing autonomous MLE baselines on most evaluated tasks and reaches expert-level performance on selected challenges, suggesting that domain-conditioned planning and artifact-level verification are important ingredients for reliable autonomous medical MLE.

This manuscript should be read as an initial technical report rather than a definitive claim about a finished system. The current version establishes the framework, reports the first benchmark evidence, and exposes the design choices needed to make agentic medical model development auditable. We will continue to improve AMID by expanding controlled backend comparisons, adding ablations for data profiling, method-lane planning, two-stage optimization, and verification gates, strengthening reviewer-style audit analyses, and evaluating cost-aware strategies for long-horizon runs. Future versions will also broaden validation across additional tasks, model-runtime combinations, resource settings, and deployment-oriented artifact checks, so that the system's performance gains, failure modes, and practical boundaries can be characterized more completely.

References

Eser Aygün, Anastasiya Belyaeva, Gheorghe Comanici, Marc Coram, Hao Cui, Jake Garrison, Renee Johnston, Anton Kast, Cory Y McLean, Peter Norgaard, et al. An AI system to help scientists write expert-level empirical software. *Nature*, pages 1–3, 2026.

Amparo Soeli Betancourt Tarifa, Faisal Mahmood, Uffe Bernchou, and Peter Jan Koopmans. Panther challenge: Public training dataset, April 2025. URL https://doi.org/10.5281/zenodo.15192302.

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Emergent autonomous scientific research capabilities of large language models. *Nature*, 624:570–578, 2023.

M Jorge Cardoso, Wenqi Li, Richard Brown, Nic Ma, Eric Kerfoot, Yiheng Wang, Benjamin Murrey, Andriy Myronenko, Can Zhao, Dong Yang, et al. MONAI: An open-source framework for deep learning in healthcare. *arXiv preprint arXiv:2211.02701*, 2022.

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

Jiefeng Chen, Bhavana Dalvi Mishra, Jaehyun Nam, Rui Meng, Tomas Pfister, and Jinsung Yoon. MARS: Modular agent with reflective search for automated AI research. *arXiv preprint arXiv:2602.02660*, 2026.

Richard J Chen, Tong Ding, Ming Y Lu, Drew FK Williamson, Guillaume Jaume, Andrew H Song, Bowen Chen, Andrew Zhang, Daniel Shao, Muhammad Shaban, et al. Towards a general-purpose foundation model for computational pathology. *Nature Medicine*, 30(3):850–862, 2024.

Audrey Cheng, Shu Liu, Melissa Pan, Zhifei Li, Bowen Wang, Alex Krentsel, Tian Xia, Mert Cemri, Jongseok Park, Shuo Yang, et al. Barbarians at the gate: How AI is upending systems research. *arXiv preprint arXiv:2510.06189*, 2025.

Alex J DeGrave, Joseph D Janizek, and Su-In Lee. AI for radiographic COVID-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, 3(7):610–619, 2021.

Shangheng Du, Xiangchao Yan, Dengyang Jiang, Jiakang Yuan, Yusong Hu, Xin Li, Liang He, Bo Zhang, and Lei Bai. AutoMLGen: Navigating fine-grained optimization for coding agents. *arXiv preprint arXiv:2510.08511*, 2025.

Shangheng Du, Xiangchao Yan, Jinxin Shi, Zongsheng Cao, Shiyang Feng, Zichen Liang, Boyuan Sun, Tianshuo Peng, Yifan Zhou, Xin Li, et al. MLEvolve: A self-evolving framework for automated machine learning algorithm discovery. *arXiv preprint arXiv:2606.06473*, 2026.

Darius A Faroughy, Sofia Palacios Schweitzer, Ian Pang, Siddharth Mishra-Sharma, and David Shih. Collider-bench: Benchmarking AI agents with particle physics analysis reproduction. *arXiv preprint arXiv:2605.13950*, 2026. doi: 10.48550/arXiv.2605.13950. URL https://arxiv.org/abs/2605.13950.

Dyke Ferber, Lars Hilgers, Christiane Höper, Benedict Kinny-Köster, Jan-Niklas Eckardt, Katharina Egger-Heidrich, Marius Bill, Martin MK Schneider, Jan Clusmann, Lejla Kadric, et al. Towards autonomous medical artificial intelligence agents. *Nature*, pages 1–10, 2026.

Alexandre Filiot, Paul Jacob, Alice Mac Kain, and Charlie Saillard. Phikon-v2, a large and public feature extractor for biomarker prediction. *arXiv preprint arXiv:2409.09173*, 2024.

Shanghua Gao, Ada Fang, and Marinka Zitnik. Autoscientists: Self-organizing agent teams for long-running scientific experimentation. *arXiv preprint arXiv:2605.28655*, 2026a.

Yifan Gao, Haoyue Li, Feng Yuan, Xin Gao, Weiran Huang, and Xiaosong Wang. Camyla: Scaling autonomous research in medical image segmentation. *arXiv preprint arXiv:2604.10696*, 2026b.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Judy Wawira Gichoya, Imon Banerjee, Ananth Reddy Bhimireddy, John L Burns, Leo Anthony Celi, Li-Ching Chen, Ramon Correa, Natalie Dullerud, Marzyeh Ghassemi, Shih-Cheng Huang, et al. AI recognition of patient race in medical imaging: a modelling study. *The Lancet Digital Health*, 4(6):e406–e414, 2022.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, et al. Towards an AI co-scientist. *arXiv preprint arXiv:2502.18864*, 2025.

Titus Griebel, Anwai Archit, and Constantin Pape. Segment anything for histopathology. *arXiv preprint arXiv:2502.00408*, 2025.

Yi Guo, Shichong Zhou, Jun Shi, and Yuanyuan Wang. Ultrasound image enhancement challenge 2023, April 2023. URL https://doi.org/10.5281/zenodo.7841250.

Ibrahim Ethem Hamamci, Sezgin Er, Enis Simsar, Atif Emre Yuksel, Sadullah Gultekin, Serife Damla Ozdemir, Kaiyuan Yang, Hongwei Bran Li, Sarthak Pati, Bernd Stadlinger, et al. Dentex: An abnormal tooth detection with dental enumeration and diagnosis benchmark for panoramic x-rays. *arXiv preprint arXiv:2305.19112*, 2023.

Keqi Han, Songlin Zhao, Yao Su, Xiang Li, Yixuan Yuan, Lifang He, and Carl Yang. Towards a virtual neuroscientist: Autonomous neuroimaging analysis via multi-agent collaboration. *arXiv preprint arXiv:2605.09366*, 2026.

Yufan He, Pengfei Guo, Yucheng Tang, Andriy Myronenko, Vishwesh Nath, Ziyue Xu, Dong Yang, Can Zhao, Benjamin Simon, Mason Belue, et al. Vista3d: A unified segmentation foundation model for 3d medical imaging. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20863–20873, 2025.

Moritz R Hernandez Petzsche, Ezequiel De La Rosa, Uta Hanning, Roland Wiest, Waldo Valenzuela, Mauricio Reyes, Maria Meyer, Sook-Lei Liew, Florian Kofler, Ivan Ezhov, et al. Isles 2022: A multi-center magnetic resonance imaging stroke lesion segmentation dataset. *Scientific data*, 9(1):762, 2022.

Fabian Hörst, Moritz Rempe, Lukas Heine, Constantin Seibold, Julius Keyl, Giulia Baldini, Selma Ugurel, Jens Siveke, Barbara Grünwald, Jan Egger, et al. CellViT: Vision transformers for precise cell segmentation and classification. *Medical Image Analysis*, 94:103143, 2024.

Chuxuan Hu, Liyun Zhang, Yeji Lim, Aum Wadhwani, Austin Peters, and Daniel Kang. REPRO-bench: Can agentic AI systems assess the reproducibility of social science research? *arXiv preprint arXiv:2507.18901*, 2025. doi: 10.48550/arXiv.2507.18901. URL https://arxiv.org/abs/2507.18901.

Kexin Huang, Serena Zhang, Hanchen Wang, Yuanhao Qu, Yingzhou Lu, Ryan Li, Yusuf Roohani, Lin Qiu, Shiyi Cao, Gavin Li, Junze Zhang, Di Yin, Rick Wierenga, Deniz Kavi, Sherry Liu, Tianwei She, Shruti Marwaha, Jennefer N. Carter, Xin Zhou, Matthew T. Wheeler, Jonathan A. Bernstein, Mengdi Wang, Peng He, Jingtian Zhou, Michael P. Snyder, Le Cong, Aviv Regev, and Jure Leskovec. Autonomous biomedical research with an artificial intelligence agent. *Science*, art. eadz4351, July 2026a. doi: 10.1126/science.adz4351. URL https://doi.org/10.1126/science.adz4351.

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

Ziyang Huang, Yi Cao, Ali K Shargh, Jing Luo, Ruidong Mei, Mohd Zaki, Zhan Liu, Wyatt Bunstine, William Jurayj, Somdatta Goswami, Tyrel McQueen, Michael Shields, Jaafar El-Awady, Paulette Clancy, Benjamin Van Durme, Nicholas Andrews, William Walden, and Daniel Khashabi. Can coding agents reproduce findings in computational materials science? *arXiv preprint arXiv:2605.00803*, 2026b. doi: 10.48550/arXiv.2605.00803. URL https://arxiv.org/abs/2605.00803.

Tal Ifargan, Lukas Hafner, Maor Kern, Ori Alcalay, and Roy Kishony. Autonomous LLM-driven research—from data to human-verifiable research papers. *NEJM AI*, 2(1):AIoa2400555, 2025.

Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2):203–211, 2021.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

Peter Jansen, Oyvind Tafjord, Marissa Radensky, Pao Siangliulue, Tom Hope, Bhavana Dalvi, Bodhisattwa Prasad Majumder, Daniel S Weld, and Peter Clark. CodeScientist: End-to-end semi-automated scientific discovery with code-based experimentation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13370–13467, 2025.

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. URL https://arxiv.org/abs/2502.13138.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Peri, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

Jiajie Jin, Yuyang Hu, Kai Qiu, Qi Dai, Chong Luo, Guanting Dong, Xiaoxi Li, Tong Zhao, Xiaolong Ma, Gongrui Zhang, et al. Toward generalist autonomous research via hypothesis-tree refinement. *arXiv preprint arXiv:2606.11926*, 2026.

Yuan Jin, Antonio Pepe, Gian Marco Melito, Yuxuan Chen, Yunsu Byeon, Hyeseong Kim, Kyungwon Kim, Doohyun Park, Euijoon Choi, Dosik Hwang, et al. Towards the automatic segmentation, modeling and meshing of the aortic vessel tree from multicenter acquisitions: An overview of the seg. a. 2023 segmentation of the aorta challenge. *arXiv preprint arXiv:2510.24009*, 2025.

Roshan Kenia, Xiaoman Zhang, and Pranav Rajpurkar. Rex-mle: The autonomous agent benchmark for medical imaging challenges. *arXiv preprint arXiv:2512.17838*, 2025.

Gyeongwon James Kim, Alex Wilf, Louis-Philippe Morency, and Daniel Fried. From reproduction to replication: Evaluating research agents with progressive code masking. *arXiv preprint arXiv:2506.19724*, 2025. doi: 10.48550/arXiv.2506.19724. URL https://arxiv.org/abs/2506.19724.

Wonkyeong Lee, Fabian Wagner, Adrian Galdran, Yongyi Shi, Wenjun Xia, Ge Wang, Xuanqin Mou, Md Atik Ahamed, Abdullah Al Zubaer Imran, Ji Eun Oh, et al. Low-dose computed tomography perceptual image quality assessment. *Medical Image Analysis*, 99:103343, 2025.

Yu Li, Chenyang Shao, Xinyang Liu, Ruotong Zhao, Peijie Liu, Hongyuan Su, Zhibin Chen, Qinglong Yang, Anjie Xu, Yi Fang, et al. Autosota: An end-to-end automated research system for state-of-the-art ai model discovery. *arXiv preprint arXiv:2604.05550*, 2026.

Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen AWM van der Laak, Bram van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, 2017.

Jiaqi Liu, Shi Qiu, Mairui Li, Bingzhou Li, Haonian Ji, Siwei Han, Xinyu Ye, Peng Xia, Zihan Dong, Congyu Zhang, et al. Autoresearchclaw: Self-reinforcing autonomous research with human-ai collaboration. *arXiv preprint arXiv:2605.20025*, 2026a.

Junqi Liu, Salena Song, Yuhan Wang, Jiawei Mao, Hardy Chen, Xiaoke Huang, Tianhao Qi, Pengfei Guo, Yucheng Tang, Yufan He, et al. Automedbench: Towards medical autoresearch with agentic ai models. *arXiv preprint arXiv:2606.01961*, 2026b.

Qianchu Liu, Sheng Zhang, Guanghui Qin, Jeya Maria Jose Valanarasu, Maximilian Rokuss, Mingyu Lu, Timothy Ossowski, Juan Manuel Zambrano Chaves, Cliff Wong, Peniel Argaw, Yashna Hasija, Mu Wei, Wen-wai Yim, Qin Liu, Zilin Jing, Jason Entenmann, Naoto Usuyama, Tristan Naumann, and Hoifung Poon. Healthagentbench: A unified benchmark suite of realistic agentic healthcare environments for challenging frontier ai agents, 2026c. URL https://arxiv.org/abs/2606.31179.

Zexi Liu, Yuzhu Cai, Xinyu Zhu, Yujie Zheng, Runkun Chen, Ying Wen, Yanfeng Wang, Siheng Chen, et al. ML-Master: Towards AI-for-AI via integration of exploration and reasoning. *arXiv preprint arXiv:2506.16499*, 2025.

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.

Bohan Lyu, Yucheng Yang, Siqiao Huang, Jiaru Zhang, Qixin Xu, Xinghan Li, Xinyang Han, Yicheng Zhang, Huaqing Zhang, Runhan Huang, Kaicheng Yang, Zitao Chen, Wentao Guo, Junlin Yang, Xinyue Ai, Wenhao Chai, Yadi Cao, Ziran Yang, Kun Wang, Dapeng Jiang, Huan-ang Gao, Shange Tang, Chengshuai Shi, Simon S Du, Max Simchowitz, Jiantao Jiao, Dawn Song, and Chi Jin. MLS-bench: A holistic and rigorous assessment of AI systems on building better AI. *arXiv preprint arXiv:2605.08678*, 2026. doi: 10.48550/arXiv.2605.08678. URL https://arxiv.org/abs/2605.08678.

Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. *Nature Communications*, 15(1):654, 2024a. doi: 10.1038/s41467-024-44824-z.

Jun Ma, Ronald Xie, Shamini Ayyadhury, Cheng Ge, Anubha Gupta, Ritu Gupta, Song Gu, Yao Zhang, Gihun Lee, Joonkee Kim, et al. The multimodality cell segmentation challenge: toward universal solutions. *Nature Methods*, 21(6):1103–1113, 2024b.

Jun Ma, Zongxin Yang, Sumin Kim, Bihui Chen, Mohammed Baharoon, Adibvafa Fallahpour, Reza Asakereh, Hongwei Lyu, and Bo Wang. MedSAM2: Segment anything in 3D medical images and videos. *arXiv preprint arXiv:2504.03600*, 2025.

Rui Meng, Bhavana Dalvi Mishra, Jiefeng Chen, Chun-Liang Li, Palash Goyal, Mihir Parmar, Yiwen Song, Yale Song, Rajarishi Sinha, Parthasarathy Ranganathan, et al. ScientistOne: Towards human-level autonomous research via chain-of-evidence. *arXiv preprint arXiv:2605.26340*, 2026.

Cathy Ong Ly, Balagopal Unnikrishnan, Tony Tadic, Tirth Patel, Joe Duhamel, Sonja Kandel, Yasbanoo Moayedi, Michael Brudno, Andrew Hope, Heather Ross, et al. Shortcut learning in medical AI hinders generalization: method for estimating AI model generalization without external data. *npj Digital Medicine*, 7(1):124, 2024.

Marius Pachitariu, Michael Rariden, and Carsen Stringer. Cellpose-sam: superhuman generalization for cellular segmentation. *bioRxiv*, pages 2025–04, 2025.

Yingming Pu, Tao Lin, and Hongyu Chen. PiFlow: Principle-aware scientific discovery with multi-agent collaboration. *arXiv preprint arXiv:2505.15047*, 2025.

Ao Qu, Han Zheng, Zijian Zhou, Yihao Yan, Yihong Tang, Shao Yong Ong, Fenglu Hong, Kaichen Zhou, Chonghe Jiang, Minwei Kong, et al. Coral: Towards autonomous multi-agent evolution for open-ended discovery. *arXiv preprint arXiv:2604.01658*, 2026.

Ben Rank, Hardik Bhatnagar, Ameya Prabhu, Shira Eisenberg, Karina Nguyen, Matthias Bethge, and Maksym Andriushchenko. PostTrainBench: Can LLM agents automate LLM post-training? *arXiv preprint arXiv:2603.08640*, 2026. doi: 10.48550/arXiv.2603.08640. URL https://arxiv.org/abs/2603.08640.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

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. *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 5977–6043, 2025.

Mark Schuiveling, Hong Liu, Daniel Eek, Gerben E Breimer, Karijn PM Suijkerbuijk, Willeke AM Blokx, and Mitko Veta. A novel dataset for nuclei and tissue segmentation in melanoma with baseline nuclei segmentation and tissue segmentation benchmarks. *GigaScience*, 14:giaf011, 2025.

Zachary S Siegel, Sayash Kapoor, Nitya Nadgir, Benedikt Stroebl, and Arvind Narayanan. CORE-bench: Fostering the credibility of published research through a computational reproducibility agent benchmark. *arXiv preprint arXiv:2409.11363*, 2024. doi: 10.48550/arXiv.2409.11363. URL https://arxiv.org/abs/2409.11363.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. PaperBench: Evaluating AI's ability to replicate AI research. *arXiv preprint arXiv:2504.01848*, 2025. doi: 10.48550/arXiv.2504.01848. URL https://arxiv.org/abs/2504.01848.

Jiabin Tang, Lianghao Xia, Zhonghang Li, and Chao Huang. AI-Researcher: Autonomous scientific innovation. *arXiv preprint arXiv:2505.18705*, 2025.

Florian Trost, Bide Zhang, Ines Aring, Marcus Bauer, Lennert Glamann, Michael Wessolly, Kyra Johnson, Heike Göbel, Tristan Lerbs, Taban Sangenne, et al. An agentic framework for autonomous scientific discovery in cancer pathology. *Nature Medicine*, pages 1–13, 2026.

Tao Tu, Shekoofeh Azizi, Danny Driess, Mike Schaekermann, Mohamed Amin, Pi-Chuan Chang, Andrew Carroll, Chuck Lau, Ryutaro Tanno, Ira Ktena, et al. Towards generalist biomedical AI. *arXiv preprint arXiv:2307.14334*, 2023.

Rejin Varghese and M Sambath. YOLOv8: A novel object detection algorithm with enhanced performance and robustness. In *2024 International conference on advances in data engineering and intelligent computing systems (ADICS)*, pages 1–6. IEEE, 2024.

Cheng Wang, Zhibin He, Zhihao Peng, Shengyuan Liu, Yufan Hu, Yang Carl, He Lifang, Lichao Sun, Xiang Li, and Yixuan Yuan. Neuroclaw technical report. *arXiv preprint arXiv:2604.24696*, 2026a.

Yubo Wang, Zhoujun Zhuang, Anirudh Srinivasa, Pradeep Dasigi, and Hannaneh Hajishirzi. SciAgent: Tool-augmented language models for scientific reasoning. *arXiv preprint arXiv:2402.11451*, 2024.

Zhen Wang, Fan Bai, Zhongyan Luo, Jinyan Su, Kaiser Sun, Xinle Yu, Jieyuan Liu, Kun Zhou, Claire Cardie, Mark Dredze, Eric P Xing, and Zhiting Hu. FIRE-bench: Evaluating agents on the rediscovery of scientific insights. *arXiv preprint arXiv:2602.02905*, 2026b. doi: 10.48550/arXiv.2602.02905. URL https://arxiv.org/abs/2602.02905.

Jakob Wasserthal, Hanns-Christian Breit, Manfred T Meyer, Maurice Pradella, Daniel Hinck, Alexander W Sauter, Tobias Heye, Daniel T Boll, Joshy Cyriac, Shan Yang, et al. Totalsegmentator: robust segmentation of 104 anatomic structures in ct images. *Radiology: Artificial Intelligence*, 5(5):e230024, 2023.

Yixuan Weng, Minjun Zhu, Qiujie Xie, Qiyao Sun, Zhen Lin, Sifan Liu, and Yue Zhang. Deepscientist: Advancing frontier-pushing scientific findings progressively. *arXiv preprint arXiv:2509.26603*, 2025.

Hongtao Wu, Boyun Zheng, Dingjie Song, Yu Jiang, Jianfeng Gao, Lei Xing, Lichao Sun, and Yixuan Yuan. Towards a medical AI scientist. *arXiv preprint arXiv:2603.28589*, 2026.

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.

Kaiyuan Yang, Fabio Musio, Yihui Ma, Norman Juchler, Johannes C Paetzold, Rami Al-Maskari, Luciano Höher, Hongwei Bran Li, Ibrahim Ethem Hamamci, Anjany Sekuboyina, et al. Benchmarking the cow with the topcow challenge: Topology-aware anatomical segmentation of the circle of willis for cta and mra. *ArXiv*, pages arXiv–2312, 2025a.

Ruofeng Yang, Yongcan Li, and Shuai Li. ARIS: Autonomous research via adversarial multi-agent collaboration. *arXiv preprint arXiv:2605.03042*, 2026.

Xu Yang, Xiao Yang, Shikai Fang, Yifei Zhang, Jian Wang, Bowen Xian, Qizheng Li, Jingyuan Li, Minrui Xu, Yuante Li, et al. R&D-Agent: An llm-agent framework towards autonomous data science. *arXiv preprint arXiv:2505.14738*, 2025b.

Yuzhe Yang, Haoran Zhang, Judy W Gichoya, Dina Katabi, and Marzyeh Ghassemi. The limits of fair medical imaging AI in real-world generalization. *Nature Medicine*, 30(10):2838–2848, 2024.

Christine Ye, Sihan Yuan, Suchetha Cooray, Steven Dillmann, Ian L V Roque, Dalya Baron, Philipp Frank, Sergio Martin-Alvarez, Nolan Koblischke, Frank J Qu, Diyi Yang, Risa Wechsler, and Ioana Ciuca. ReplicationBench: Can AI agents replicate astrophysics research papers? *arXiv preprint arXiv:2510.24591*, 2025. doi: 10.48550/arXiv.2510.24591. URL https://arxiv.org/abs/2510.24591.

Kai Zhang, Rong Zhou, Eashan Adhikarla, Zhiling Yan, Yixin Liu, Jun Yu, Zhengliang Liu, Xun Chen, Brian D Davison, Hui Ren, et al. BiomedGPT: A generalist vision-language foundation model for diverse biomedical tasks. *Nature Medicine*, 2024. doi: 10.1038/s41591-024-03185-2.

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*, 2026.

Theodore Zhao, Yu Gu, Jianwei Yang, Naoto Usuyama, Ho Hin Lee, Sid Kiblawi, Tristan Naumann, Jianfeng Gao, Angela Crabtree, Jacob Abel, et al. A foundation model for joint segmentation, detection and recognition of biomedical objects across nine modalities. *Nature Methods*, 22(1):166–176, 2025.

Zihao Zhao, Frederik Hauke, Juliana De Castilhos, Jakob Nikolas Kather, Sven Nebelung, and Daniel Truhn. From clinical intent to clinical model: An autonomous coding-agent framework for clinician-driven ai development. *arXiv preprint arXiv:2604.17110*, 2026.

Hong-Yu Zhou, Julián Nicolás Acosta, Subathra Adithan, Suvrankar Datta, Eric J Topol, and Pranav Rajpurkar. MedVersa: a generalist foundation model for diverse medical imaging tasks. *NEJM AI*, 3(4):AIoa2500595, 2026.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.