BCER Agent: Reliable Long-Horizon MRI Workflow Execution via Compilation, Artifact Binding, and Bounded Local Recovery

Ziyang Long[1,2], Xinqi Li[1,3], Junzhou Chen[1], Yifan Gao[1,4], Debiao Li[1,2], and Hsin-Jung Yang[1]

[1] Biomedical Imaging Research Institute, Cedars-Sinai Medical Center, Los Angeles, CA, USA
{Junzhou.Chen,Yifan.Gao}@cshs.org; {Debiao.Li,hsin-jung.yang}@csmc.edu
[2] Department of Bioengineering, University of California, Los Angeles, CA, USA
ziyanglong@ucla.edu
[3] Berlin Ultrahigh Field Facility, Max Delbrück Center for Molecular Medicine in the Helmholtz Association, Berlin, Germany
xinqi.li@mdc-berlin.de
[4] Tsinghua Medicine, Tsinghua University, Beijing, China

Abstract. Recent medical VLM and agent studies are often evaluated on 2D images or short tool-calling tasks, while practical MRI analysis requires long, interdependent workflows over 3D/4D volumetric data. In such workflows, reactive tool-calling agents are vulnerable to cascading failures caused by incorrect intermediate references, incompatible tool arguments, and weak control over cross-step dependencies. We present **BCER** (Brain–Cerebellum–Extremity–Reflector), a controller architecture for reliable long-horizon MRI workflow execution. BCER separates high-level planning from execution and supports bounded local recovery. We evaluate BCER on a multi-organ MRI benchmark spanning brain, prostate, and cardiac tasks with both short- and long-chain workflows, under matched task contracts across controller variants and multiple backbone models. Compared with reactive baselines, BCER consistently improves end-to-end execution, with the largest gains on long-chain workflows. BCER also supports auditability by preserving explicit links between final outputs and intermediate artifacts and measurements. Code and benchmark are available at https://github.com/Albertlongzi/BCER.

Keywords: Medical Agent · MRI · Tool orchestration · Auditability.

1 Introduction

Recent medical VLM and agent studies have shown promising progress in medical image understanding and tool-augmented reasoning [19, 7, 5, 24]. However, many existing evaluations remain centered on relatively bounded settings, such

as pre-selected image inputs, slice- or case-level interpretation [8, 13], visual question answering [6, 9], or short tool-use sequences [13]. In clinical practice, however, MRI analysis is often better framed as a multi-stage workflow than as a one-hop question-answering task. It may begin upstream (including reconstruction in some settings), span multiple dependent steps, and require consistent handling of intermediate outputs across tools.

This creates two practical requirements for deployable medical agents that remain underexplored: (i) reliable execution over intermediate artifacts and cross-step dependencies, and (ii) auditability, i.e., linking final conclusions to intermediate evidence and measurements [14].

Many current tool-augmented baselines and agent frameworks adopt reactive, interleaved planning-and-acting loops (e.g., ReAct-style prompting [25, 17, 19]). While effective for flexible short-form interactions, these designs offer limited explicit control over intermediate artifact dependencies [22], which becomes increasingly important in long MRI workflows [4].

In this work, we study an *execution gap* in long-horizon MRI agent workflows: failures caused by incorrect intermediate-result references, incompatible tool arguments, or broken step dependencies can accumulate and prevent task completion, even when individual tools are accurate. We therefore treat *execution reliability*—including end-to-end completion and robust handling of execution failures—as a first-class objective.

To address this, we propose **BCER** (Brain–Cerebellum–Extremity–Reflector), a controller architecture for reliable long-horizon MRI workflow execution that separates high-level planning from constrained execution and bounded local recovery, enabling end-to-end workflows whose intermediate artifacts remain traceable for downstream review.

Our main contributions are three-fold:

- **A workflow-oriented MRI agent setting for multi-organ research pipelines.** We formulate medical agents for *multi-step MRI workflows* beyond diagnosis QA, targeting research-oriented pipelines across organs and tasks, including workflows that can start from raw data when available.
- **BCER: plan–execution separation with bounded local recovery.** We propose the Brain–Cerebellum–Extremity–Reflector (BCER) architecture, which separates goal-level planning from constrained execution and supports localized repair-and-retry instead of full workflow restarts, improving robustness in long MRI tool chains while preserving traceable intermediate artifacts.
- **A task-contract benchmark for short- and long-chain MRI workflows.** We introduce a task-contract-based evaluation that measures both task success and completion fidelity across controller variants and backbone models.

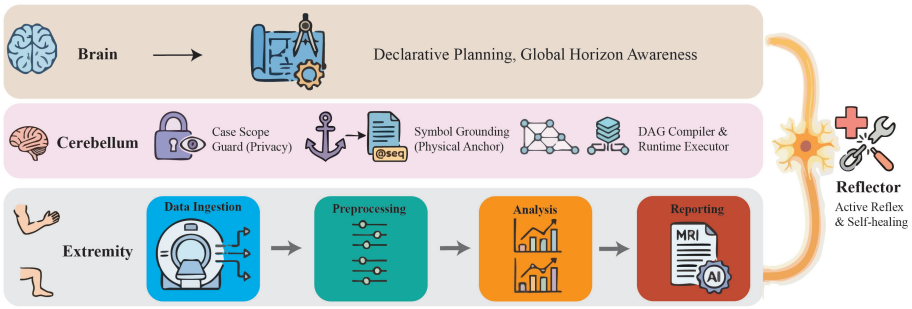

Fig. 1. BCER overview. The Brain generates a constrained plan sketch from the user goal and available MRI inputs. The Cerebellum compiles the sketch into an executable workflow graph and executes it under run-time constraints, dispatching tools from the Extremity (MRI tool library) while recording intermediate artifacts and outcomes. The Reflector performs bounded step- or sub-workflow repair when failures occur.

2 Method

2.1 Problem Setup

We consider an MRI agent task in which the system receives volumetric imaging data V, accessible metadata M, and a natural-language user goal g:

$$I = (V, M, g). \tag{1}$$

The goal is not only to produce a final answer, but to complete a *multi-step MRI workflow* and return the required task deliverables (e.g., processed volumes, masks, measurements, classifications, or a structured report), together with execution records for traceability.

We define success by *task-contract satisfaction*: all required workflow steps and outputs specified by the task are completed under the system's execution constraints. This formulation supports evaluation of both end-to-end task completion and step-level completion in tool chains.

2.2 BCER

Brain. The Brain generates a *plan sketch* from (V, M, g), describing intended workflow steps and constraints, but not a directly executable workflow. This separation of conceptual planning from low-level execution draws inspiration from prior modular agent systems [23, 16], but is adapted here to the requirements of MRI workflows. In particular, the Brain does not execute tools, does not directly manipulate file paths, and may leave some executable fields to be completed downstream.

Extremity. The Extremity is BCER's executable tool layer. It provides standardized MRI-facing tools for data I/O, preprocessing, analysis, quantification,

and reporting, with uniform argument schemas, typed outputs, and normalized error codes for runtime dispatch and recovery. The library includes both wrapped conventional operators (e.g., reconstruction, denoising, registration) and integrated task-specific modules drawn from established MRI subproblems (e.g., segmentation, detection, and correction) [1, 15, 10, 26, 12], as well as workflow utilities for intermediate-state resolution (such as ED/ES frame selection) and evidence-backed decision tools that combine measurements with explicit rules/skills for downstream classification and report generation.

Cerebellum. The Cerebellum is the runtime controller that converts the Brain's high-level plan sketch into an executable workflow. The sketch is not executed directly because it may be partially specified and may omit dispatch-level details. Instead, the Cerebellum validates the proposed steps against task-level interface and dependency constraints, resolves only dispatch-required defaults or unambiguous links, and produces a runnable workflow graph. In this sense, compilation does not copy a fixed task template; rather, it turns a partially specified plan into an executable form with explicit dependencies and bounded execution semantics.

Before each tool call, the Cerebellum resolves references to inputs and intermediate results via symbolic tokens: `@seq.*`, `@node.<id>.<field>`, `@case.*`, and `@runtime.*`. It binds these tokens to concrete values/paths within the current case run so tools consume actual artifacts rather than LLM-generated strings. Drawing inspiration from artifact-passing mechanisms in modular tool-use agents [16], this explicit symbolic binding creates a more robust dataflow interface for large medical artifacts. This *binding* is a key distinction from direct ReAct-style tool calling and is isolated empirically through the *ReAct+Bind* controller variant.

The Cerebellum executes nodes when prerequisites are satisfied, updates per-case state and an event trace with status/outputs/errors, and enforces run-time constraints (e.g., case-level scope/sandbox boundaries). Thus, failures surface as explicit node failures with traceable context rather than silently corrupting downstream steps.

Reflector. The Reflector handles failures of required nodes at runtime via *bounded local repair*. Prior reflection-style mechanisms in LLM agents often operate at the level of full-turn or broad response revision [18, 11], which becomes increasingly expensive in long MRI workflows. Instead of restarting the workflow, it inspects the failed node, its arguments, the error message, and available artifacts, then proposes a minimal retry patch (e.g., fixing a token reference, removing an invalid override, or applying a safe argument adjustment). The Cerebellum then retries only the failed node (or a small affected sub-workflow) under the same run-time constraints while preserving successful upstream results. Repair follows a two-stage strategy: deterministic fixes are attempted first; a constrained LLM-based repair is invoked only if needed. This improves robustness in long tool chains while keeping recovery localized and auditable.

BCER is a domain-structured integration of planning, compilation, constrained execution, and local recovery for MRI workflows, rather than a claim

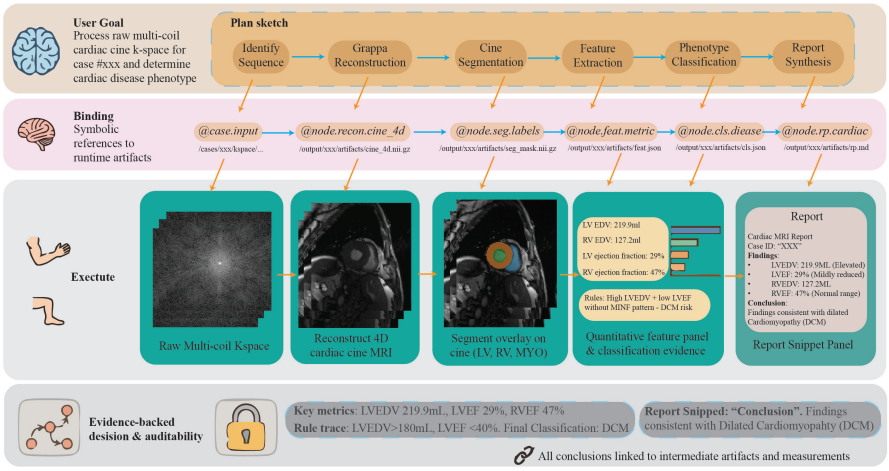

Fig. 2. BCER end-to-end cardiac workflow walkthrough. From top to bottom, the figure illustrates four linked layers of execution. **Top:** a plan sketch generated from the user goal, shown here as a multi-step cardiac pipeline from sequence identification and reconstruction to segmentation, feature extraction, phenotype classification, and report synthesis. **Second row:** symbolic artifact binding, where abstract references (e.g., `@case.input`, `@node.*`) are resolved to concrete runtime outputs. **Third row:** execution outputs produced by the tool chain, including reconstructed cine MRI, segmentation overlays, quantitative measurements, and the report snippet. **Bottom:** evidence-backed traceability, where final conclusions remain linked to intermediate artifacts, measurements, and rule traces for downstream review.

that each component is independently novel. Unlike open-ended symbolic planning or directly executable LLM plans, the Brain produces a constrained sketch that the Cerebellum compiles into a typed, case-scoped workflow graph with dependency resolution and symbolic artifact binding. Figure 2 provides an end-to-end cardiac walkthrough, illustrating how BCER transforms a user goal into a compiled multi-step workflow with staged execution outputs and evidence-linked reporting.

3 Experiments and Results

3.1 Benchmark Setup

We use four public MRI datasets spanning brain, prostate, and cardiac domains: BraTS 2018 [2], fastMRI Prostate [20], ACDC [3], and CMRxRecon [21]. The benchmark contains 8 tasks covering denoising, super-resolution, segmentation, reconstruction, registration, grading/classification, and end-to-end report generation (Table 1). Each run is defined by a task contract specifying the input case, goal, allowed tool subset, and required outputs (gold deliverables). The same

Table 1. Benchmark task suite overview. Tasks are defined by task contracts and span both short-chain (S) and long-chain (L) MRI workflows across brain, prostate, and cardiac domains. The table summarizes the input type, organ domain, task-specific case count, and required gold output for each task.

Task	Chain	Input	Organ	Cases	Gold Output
Denoise	S	NIfTI	brain prostate	200	denoised volume
SuperRes	S	NIfTI	multi-organ	310	super-resolved volume
Segment	S	DICOM NIfTI	brain	100	organ mask
Recon	S	k-space	cardiac	10	reconstructed image volume
Register	S	DICOM NIfTI	prostate	100	aligned volume
BrainGrade	L	NIfTI	brain	100	grade/class label
ProstateRpt	L	DICOM	prostate	100	structured report
CardiacRpt	L	k-space DICOM	cardiac	110	structured report
Total	–	–	–	1030	–

contracts are used across controller arms, yielding matched action spaces and success criteria.

We group tasks into *short-chain* (single- or few-step imaging operations) and *long-chain* (multi-stage analysis and reporting workflows), and use this split throughout the results. We evaluate four controller variants in end-to-end execution: **ReAct**, **ReAct+Bind**, **ReAct+Bind+Ref**, and **BCER**. These variants are structured as a progressive ablation ladder: starting from direct reactive tool calling, then adding symbolic artifact binding, then local repair, and finally BCER's full plan–execution control stack. This setup enables both comparison against a standard reactive baseline and attribution of the gains from each added controller component. We first compare these controllers on the benchmark tasks under normal execution conditions, and then examine whether the same trends hold across multiple backbone models.

For metrics, we report **Success Rate (SR)** and **Task Completion Rate (TCR)** against fixed task contracts specifying inputs, goals, allowed tools, required milestones, and deliverables. SR is binary per case: all required milestones and deliverables must be produced and validated. TCR is the fraction of reference contract milestones validated, rather than attempted or generated steps; arbitrary files or free-form answers do not count. These metrics measure controller-level execution reliability, not clinical diagnostic accuracy or standalone tool performance.

3.2 End-to-end controller comparison

Table 2 compares task-level SR/TCR for four controller variants using Qwen3-VL-30B-A3B-Thinking as the backbone. The main trend is not only that per-

Table 2. Task-level success rate (SR, %) / task completion rate (TCR, %) for end-to-end controller comparison under normal execution. Tasks are grouped into short-chain and long-chain workflows.

Task	ReAct	ReAct+Bind	ReAct+Bind+Ref	BCER
Short-chain tasks				
Denoise	95/95	98/98	100/100	**100/100**
Super-resolution	75/75	90/90	95/95	**100/100**
Segmentation	89/91	98/99	100/100	**100/100**
Reconstruction	100/100	100/100	100/100	**100/100**
Registration	69/77	100/100	100/100	**100/100**
Long-chain tasks				
Brain grading	70/82	100/100	100/100	**100/100**
Prostate report	0/46	0/72	0/72	**99/99**
Cardiac report	0/22	63/66	88/89	**93/93**
Total	65/74	83/90	87/95	**99/99**

formance improves from ReAct to BCER, but that the dominant failure mode shifts at each stage of the controller stack.

In ReAct, failures are dominated by unstable step execution, including parameter/schema mismatches and path or input-selection errors. On long-chain tasks, the issue is often not a single missing step, but that one unstable tool call breaks downstream dependencies and causes the entire workflow to collapse.

Adding binding (*ReAct+Bind*) substantially reduces low-level intermediate-reference and path-resolution errors, which explains the large gain in overall SR/TCR (from **65/74** to **83/90**). However, many remaining failures are no longer simple reference-grounding mistakes; instead, they arise from workflow-level omissions, such as missing a required tool step or producing an incomplete processing chain.

Adding the Reflector (*ReAct+Bind+Ref*) further suppresses low-level execution errors through local repair, and improves overall performance to **87/95**. However, it still does not fully resolve failures caused by incomplete workflow structure. In our error review, these remaining cases are dominated by missing or insufficiently specified steps, together with occasional task-contract misses (e.g., incomplete completion of required resampling in a small number of short super-resolution cases).

By contrast, BCER removes most of these structural failure modes earlier, at the planning-and-compilation stage. Compared with *ReAct+Bind+Ref*, BCER generates more complete workflow graphs and more consistent step dependencies, so failures are less likely to arise from hallucinated parameters, invalid paths, or omitted nodes. As a result, the remaining errors are concentrated near the robustness limits of a small number of tools or task contracts, rather than cascading controller failures. This difference is most visible on long-chain reporting tasks: while *ReAct* fails almost completely on `ProstateRpt` and `CardiacRpt`

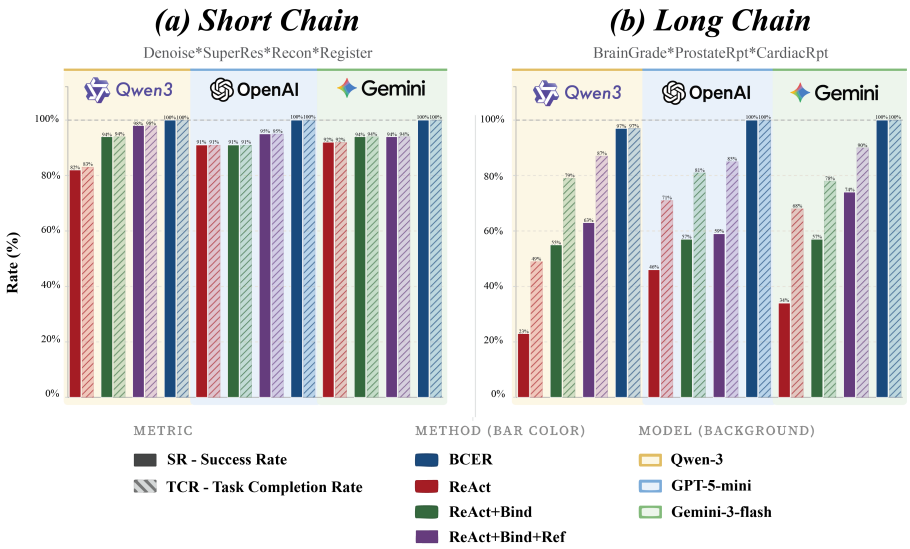

Fig. 3. Bars report SR (solid) and TCR (hatched) for **ReAct**, **ReAct+Bind**, **ReAct+Bind+Ref**, and **BCER**, aggregated over short-chain tasks (`Denoise`, `SuperRes`, `Recon`, `Register`) and long-chain tasks (`BrainGrade`, `ProstateRpt`, `CardiacRpt`).

(**0**/**46** and **0**/**22**), BCER reaches **99**/**99** and **93**/**93**, indicating that compiled execution improves not only reference grounding, but overall workflow integrity.

3.3 Across Backbone Models

Figure 3 extends the controller comparison across multiple LLM backbones. On short-chain tasks, all backbones perform relatively well and the gap between methods is modest. On long-chain tasks, however, the gap widens substantially: plain *ReAct* degrades across all three backbones, and stronger base models only partially mitigate this drop. In contrast, adding *Bind* and then *Reflector* yields consistent gains for every backbone, indicating that the main bottleneck is not only model capacity, but also execution control over intermediate artifacts and workflow dependencies. Across all evaluated backbones, BCER remains the most stable variant and achieves near-saturated performance on long chains, showing that the benefits of compiled execution generalize beyond any single base model.

4 Conclusion

We presented **BCER**, a workflow controller architecture for long, multi-step MRI analysis that combines plan–execution separation, symbolic artifact binding, and bounded local recovery. Across a multi-organ MRI benchmark with short- and long-chain tasks, BCER consistently outperformed reactive tool-calling baselines, with the largest gains on long-chain workflows.

This work focuses on *controller capability* rather than the upper-bound performance of individual tools or backbone models. Under matched task contracts and tool availability, BCER improves executability by imposing stronger execution structure, at the cost of reduced flexibility. This is not a prospective clinical validation study and does not claim to replace expert-generated reports or clinician judgment; expert knowledge enters through task definitions, public benchmark annotations when available, predefined deliverables, and rule/tool interfaces. Its current design also assumes a curated tool library with explicit interfaces, rather than open-ended tool discovery. Future work will explore safer tool expansion, clinician-in-the-loop validation, and broader evaluation in prospective clinical research workflows.

BCER is a practical step toward more reliable, evidence-linked medical imaging agents: it improves end-to-end workflow robustness while preserving traceability from final outputs to intermediate artifacts and measurements, which is important for reproducibility, auditability, and eventual translational use in MRI-centered clinical research.

Acknowledgments. This work was supported in part by the National Institutes of Health under grants 1R01HL181091 and 5R01HL165211.

Disclosure of Interests. The authors have no competing interests to declare.

References

1. Adams, L.C., Makowski, M.R., Engel, G., Rattunde, M., Busch, F., Asbach, P., Niehues, S.M., Vinayahalingam, S., van Ginneken, B., Litjens, G., et al.: Prostate158-an expert-annotated 3t mri dataset and algorithm for prostate cancer detection. Computers in biology and medicine **148**, 105817 (2022)
2. Bakas, S., Reyes, M., Jakab, A., Bauer, S., Rempfler, M., Crimi, A., Shinohara, R.T., Berger, C., Ha, S.M., Rozycki, M., et al.: Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge. arXiv preprint arXiv:1811.02629 (2018)
3. Bernard, O., Lalande, A., Zotti, C., Cervenansky, F., Yang, X., Heng, P.A., Cetin, I., Lekadir, K., Camara, O., Ballester, M.A.G., et al.: Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: is the problem solved? IEEE transactions on medical imaging **37**(11), 2514–2525 (2018)
4. Gorgolewski, K., Burns, C.D., Madison, C., Clark, D., Halchenko, Y.O., Waskom, M.L., Ghosh, S.S.: Nipype: A flexible, lightweight and extensible neuroimaging data processing framework in python. Frontiers in Neuroinformatics **5**, 13 (2011). https://doi.org/10.3389/fninf.2011.00013
5. Kim, Y., Park, C., Jeong, H., Chan, Y.S., Xu, X., McDuff, D., Lee, H., Ghassemi, M., Breazeal, C., Park, H.W.: Mdagents: An adaptive collaboration of llms for medical decision-making. Advances in Neural Information Processing Systems **37**, 79410–79452 (2024)
6. Lau, J.J., Gayen, S., Ben Abacha, A., Demner-Fushman, D.: A dataset of clinically generated visual questions and answers about radiology images. Scientific Data **5**, 180251 (2018). https://doi.org/10.1038/sdata.2018.251

7. Li, B., Yan, T., Pan, Y., Luo, J., Ji, R., Ding, J., Xu, Z., Liu, S., Dong, H., Lin, Z., et al.: Mmedagent: Learning to use medical tools with multi-modal agent. In: Findings of the Association for Computational Linguistics: EMNLP 2024. pp. 8745–8760 (2024)
8. Li, C., Wong, C., Zhang, S., Usuyama, N., Liu, H., Yang, J., Naumann, T., Poon, H., Gao, J.: Llava-med: Training a large language-and-vision assistant for biomedicine in one day (2023). https://doi.org/10.48550/arXiv.2306.00890, https://arxiv.org/abs/2306.00890
9. Liu, B., Zhan, L.M., Xu, L., Ma, L., Yang, Y., Wu, X.M.: Slake: A semantically-labeled knowledge-enhanced dataset for medical visual question answering (2021). https://doi.org/10.48550/arXiv.2102.09542, https://arxiv.org/abs/2102.09542
10. Long, Z., Nader, B., Wang, L., Malaji, A.V., Yang, C.C., Sun, H., Saouaf, R., Daskivich, T., Kim, H., Xie, Y., et al.: Let distortion guide restoration (dgr): A physics-informed learning framework for prostate diffusion mri. arXiv preprint arXiv:2601.00226 (2026)
11. Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Qian, Y., Choi, Y.: Self-refine: Iterative refinement with self-feedback (2023). https://doi.org/10.48550/arXiv.2303.17651, https://arxiv.org/abs/2303.17651
12. Myronenko, A.: 3d mri brain tumor segmentation using autoencoder regularization. In: International MICCAI brainlesion workshop. pp. 311–320. Springer (2018)
13. Nath, V., Li, W., Yang, D., Myronenko, A., Zheng, M., Lu, Y., Liu, Z., Yin, H., Law, Y.M., Tang, Y., et al.: Vila-m3: Enhancing vision-language models with medical expert knowledge. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 14788–14798 (2025)
14. Rajpurkar, P., Chen, E., Banerjee, O., Topol, E.J.: Ai in health and medicine. Nature Medicine **28**(1), 31–38 (2022). https://doi.org/10.1038/s41591-021-01614-0
15. Saha, A., Hosseinzadeh, M., Huisman, H.: End-to-end prostate cancer detection in bpmri via 3d cnns: Effects of attention mechanisms, clinical priori and decoupled false positive reduction. Medical image analysis **73**, 102155 (2021)
16. Shen, Y., Song, K., Tan, X., Li, D., Lu, W., Zhuang, Y.: Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face (2023). https://doi.org/10.48550/arXiv.2303.17580, https://arxiv.org/abs/2303.17580
17. Shinn, N., Cassano, F., Berman, E., Gopinath, A., Narasimhan, K., Yao, S.: Reflexion: Language agents with verbal reinforcement learning, 2023. URL https://arxiv.org/abs/2303.11366 **8** (2024)
18. Shinn, N., Labash, B., Gopinath, A., Narasimhan, K., Yao, S.: Reflexion: Language agents with verbal reinforcement learning (2023). https://doi.org/10.48550/arXiv.2303.11366, https://arxiv.org/abs/2303.11366
19. Tang, X., Zou, A., Zhang, Z., Li, Z., Zhao, Y., Zhang, X., Cohan, A., Gerstein, M.: Medagents: Large language models as collaborators for zero-shot medical reasoning. In: Findings of the Association for Computational Linguistics: ACL 2024. pp. 599–621. Association for Computational Linguistics, Bangkok, Thailand (2024). https://doi.org/10.18653/v1/2024.findings-acl.33
20. Tibrewala, R., Dutt, T., Tong, A., Ginocchio, L., Lattanzi, R., Keerthivasan, M.B., Baete, S.H., Chopra, S., Lui, Y.W., Sodickson, D.K., et al.: Fastmri prostate: A public, biparametric mri dataset to advance machine learning for prostate cancer imaging. Scientific data **11**(1), 404 (2024)

21. Wang, C., Lyu, J., Wang, S., Qin, C., Guo, K., Zhang, X., Yu, X., Li, Y., Wang, F., Jin, J., et al.: Cmrxrecon: A publicly available k-space dataset and benchmark to advance deep learning for cardiac mri. Scientific Data **11**(1), 687 (2024)
22. Wang, X., Chen, Y., Yuan, L., Zhang, Y., Li, Y., Peng, H., Ji, H.: Executable code actions elicit better llm agents. In: Proceedings of the 41st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 235, pp. 50208–50232 (2024). https://doi.org/10.48550/arXiv.2402.01030, https://proceedings.mlr.press/v235/wang24h.html
23. Wang, Z., Cai, S., Liu, A., Ma, X., Liang, Y.: Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents (2023). https://doi.org/10.48550/arXiv.2302.01560, https://arxiv.org/abs/2302.01560
24. Wang, Z., Wu, J., Cai, L., Low, C.H., Yang, X., Li, Q., Jin, Y.: Medagent-pro: Towards evidence-based multi-modal medical diagnosis via reasoning agentic workflow (2025). https://doi.org/10.48550/arXiv.2503.18968, https://arxiv.org/abs/2503.18968
25. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K.R., Cao, Y.: React: Synergizing reasoning and acting in language models. In: The eleventh international conference on learning representations (2022)
26. Zhao, Y., Kellman, P., Xue, H., Yang, T., Zhang, Y., Han, Y., Simonetti, O., Tao, Q.: Reverse imaging for wide-spectrum generalization of cardiac mri segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 555–565. Springer (2025)