CT-AGENT: AN EVIDENCE-GROUNDED MULTI-TOOL AGENT FOR HETEROGENEOUS CT ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Clinical CT interpretation comprises a sequence of evidence-gathering steps. A radiologist moves across slices, localizes findings, measures lesions, compares examinations, and reconciles these observations before producing a report. Current 3D vision-language models usually compress this process into a label, answer, or generated report. Specialist imaging models expose useful intermediate results and usually operate as separate tools. We present CT-Agent, an evidence-grounded agent that organizes image-only, text-only, and image-text requests as structured tasks, invokes expert models through the Model Context Protocol, and retains masks, bounding boxes, measurements, text records, and execution provenance in a case-level evidence memory. The same evidence supports quantitative reasoning and output verification. Each capability is evaluated with a task-specific benchmark. On LiTS, CT-Agent improves mean Dice from 0.715 to 0.731 and reduces measurement errors relative to 3DMedAgent. On 40 BronAtlas cases, coordinating ATM22 and AeroPath raises Dice to 0.750 and precision to 0.993. For abdominal screening, organ-level inputs, a lightweight adapter, and calibrated thresholds recover the zero sensitivity observed under direct CT-CLIP transfer. In a 40-case CT-RATE workflow, every case completes tool execution and evidence export; verification corrects four report assertions and introduces one error. These results characterize the effects of tool orchestration and the current limits of calibration, sample size, and automatic verification.

1 INTRODUCTION

A CT examination contains more information than a single model output can conveniently represent. Anatomical structures extend across slices, lesions must be interpreted under appropriate window settings, and clinically relevant findings often depend on their location, size, attenuation, and change over time. Radiologists therefore reach a conclusion by constructing an evidence chain: they identify the target anatomy, localize suspicious findings, obtain measurements, compare prior examinations when available, and translate the assembled evidence into a report. Systems that expose these intermediate results can support the full interpretation process and make each conclusion traceable to its supporting evidence.

Recent CT foundation models and medical vision-language models have improved volumetric representation learning, visual question answering, disease prediction, and report generation (Hamamci et al., 2024; 2026; Bai et al., 2024; Li et al., 2025; Wu et al., 2025). Their usual interface maps a volume or an image-text pair directly to a final response. The output often omits the slices, masks, measurements, and decision criteria supporting that response. These missing records limit inspection when a finding is subtle, when multiple abnormalities coexist, or when a quantitative value affects the conclusion.

Specialist imaging models provide spatially precise and quantitatively useful results for detection, segmentation, reconstruction, and text processing (Isensee et al., 2021; Ma et al., 2024; Wasserthal et al., 2023). Each tool has its own anatomy, input format, output schema, and operating assumptions. Fixed pipelines handle predefined tasks and data configurations. Changing requests and conflicting outputs require adaptive tool selection and case-level state management.

Medical imaging agents have begun to connect language-model planning with external tools. Prior systems use anatomy-specific models for CT question answering, accumulate tool results in shared

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

memory, record report-generation trajectories, organize multiple agents, or retrieve supporting evidence (Mao et al., 2025; Wang et al., 2026; Roschewitz et al., 2026; Lin et al., 2026; Qi et al., 2026). A remaining design problem concerns the execution layer: it must represent heterogeneous CT requests, preserve tool outputs as source-linked case evidence, and keep the final answer within the available evidence.

We address this problem with CT-Agent, shown in Figure 1. The system first converts the request and available case data into a structured task object. An LLM controller selects expert tools from a registry and invokes them through the Model Context Protocol (MCP) (Anthropic, 2024). CT-Agent stores the returned visual, quantitative, textual, and execution records in evidence memory. The controller uses these records to choose a subsequent tool, compute case-level quantities, or revise an incomplete result. Before release, a verification step compares the candidate output with the stored evidence and either accepts it, marks uncertainty, or returns the case to the execution loop.

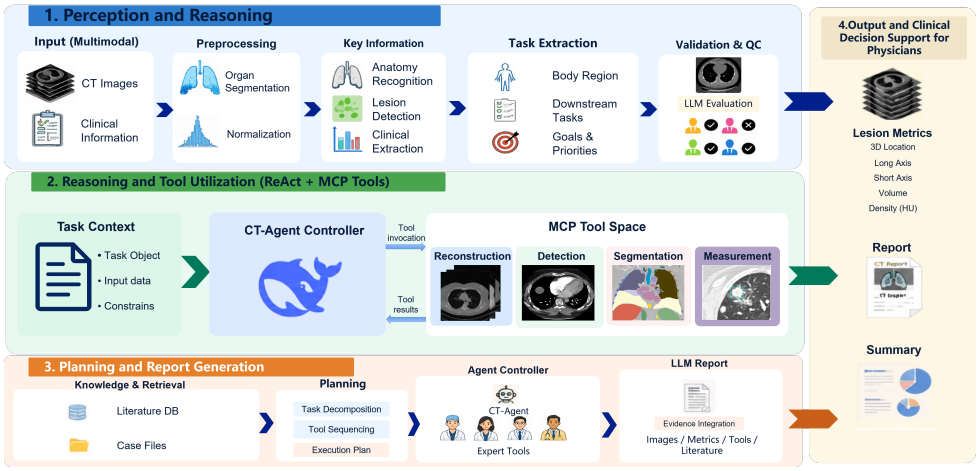


Figure 1: Overall architecture of CT-Agent. The controller converts heterogeneous case requests into executable tasks, invokes expert tools through MCP, retains their outputs as case evidence, and verifies the final response.

Figure 1 presents the controller as the connection among task interpretation, expert execution, evidence retention, and final verification. The return path from verification to the controller sends unsupported results into another evidence-gathering step before final output.

The evaluation is organized around five questions. Can the agent improve segmentation and measurement relative to an existing 3D medical agent? Can it combine complementary specialist tools more effectively than either tool alone? Can task-aware inputs and calibration make a foundation model useful outside its original anatomical setting? Can retained evidence expose conflicts in a candidate output and support a traceable correction? Finally, can the assembled system complete a case-level CT workflow? We study these questions on LiTS, BronAtlas, abdominal CT tasks, and CT-RATE. A supplementary CheXpert experiment isolates structured ReAct observation handling on radiographs. Each task retains its own metrics to preserve the corresponding failure modes.

Our contribution is an execution framework that links task representation, tool use, retained evidence, and verification within one case state. It supports heterogeneous image and text requests across expert models with different architectures. The accompanying evaluation separates agent-level coordination from the performance of the underlying tools and tests whether retained evidence can expose cross-tool conflicts and revise candidate outputs.

2 Related Work

3D medical foundation models. Volumetric vision-language models extend image-text learning to 3D scans through volume encoders, slice-sequence representations, and cross-modal alignment. CT2Rep generates radiology reports from 3D CT, M3D supports multimodal 3D medical tasks, and later generalist models broaden disease prediction, question answering, and report generation

(Hamamci et al., 2024; Bai et al., 2024; Hamamci et al., 2026; Wu et al., 2025). Brain CT reporting has also been studied with anatomy-focused multimodal systems (Li et al., 2025). These models learn broad representations. Multi-stage clinical workflows also require explicit masks, measurements, provenance, and cross-tool checks.

Specialist models and fixed pipelines. Task-specific models remain the usual choice when spatial precision or quantitative output matters. nnU-Net, TotalSegmentator, MedSAM, and VISTA3D illustrate the strength of specialized or promptable segmentation systems (Isensee et al., 2021; Wasserthal et al., 2023; Ma et al., 2024; He et al., 2025). Comparable tools exist for airway analysis, pulmonary nodules, hemorrhage, and report text. Their outputs support measurement and inspection. Integration usually relies on manually written pipelines with a fixed tool order and limited records of tool selection, conflict resolution, and evidence support.

Tool-using medical agents. Recent agents use language-model planning to organize tool execution. The CT-Agent of Mao et al. (2025) uses anatomy-specific tools for 3D CT question answering. 3DMedAgent stores tool outputs in shared memory for perception and reasoning (Wang et al., 2026). RadAgent records stepwise chest CT interpretation, MARCH uses a multi-agent hierarchy to revise reports, and EviAgent combines visual experts with retrieval for evidence-supported generation (Roschewitz et al., 2026; Lin et al., 2026; Qi et al., 2026). Our framework uses a persistent case state as the unit of integration. This state links the user request, every tool call, spatial and quantitative outputs, and the final verification decision across anatomies, task types, and output formats.

3 METHOD

3.1 TASK REPRESENTATION AND TOOL REGISTRY

CT-Agent receives image-only, text-only, or image-text input together with a user request. An input adapter converts these materials into a structured task object containing the case identifier, input modality, anatomy, requested task, available priors, constraints, and expected output. Image records retain the volume path and spatial metadata; text records retain report content, clinical questions, and timepoints. The system initializes a persistent case state from this task object. The case state contains the original task, accumulated tool observations, current completion status, and any unresolved requirements. This representation separates task intent from the file formats used by individual tools.

Each expert tool is registered with its supported anatomy and tasks, required arguments, output schema, and execution endpoint. The controller, implemented with DeepSeek-V4-Pro (DeepSeek-AI, 2026) in the current system, matches the structured task object against this registry and calls selected tools through MCP (Anthropic, 2024). Detection, segmentation, reconstruction, measurement, text extraction, and report-generation models can therefore participate in one workflow without sharing model architecture or deployment code.

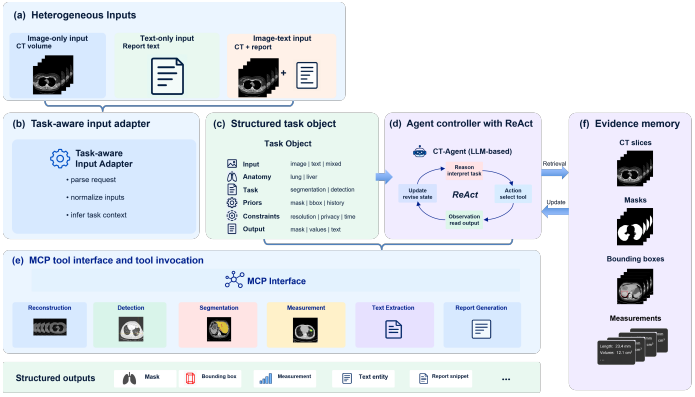


Figure 2: Task representation and tool invocation through a shared case state.

Figure 2 shows how heterogeneous inputs enter the same execution path. The task object standardizes intent before tool selection, and the MCP schema standardizes execution. This separation preserves the representation of the clinical request across different tool choices.

Table 1: Task-aware input adaptation and tool-interface mapping.

Input setting	Main fields in the task object	Compatible tool types	Typical outputs
Image-only	CT path, spatial metadata, target region, task type, available priors	Detection, segmentation, measurement, reconstruction, quality control	BBox, mask, volume, diameter, reconstructed image
Text-only	Report text, clinical question, fields to be extracted, timepoint	Information extraction, negation detection, longitudinal text comparison, report generation	Entities, measurements, negation status, change summary
Image-text	CT path, report text, clinical information, intermediate results	Image-text reasoning, report generation, differential diagnosis, key finding alert	Structured report, diagnostic suggestion, evidence summary

Table 1 makes the adapter's role concrete. The three input settings preserve different source fields and translate them into fields matched against the same tool registry. The table also defines the output types that the controller uses to decide whether a task is complete.

3.2 Planning and evidence state

The controller follows a ReAct loop (Yao et al., 2023). At each iteration, it reads the structured task object together with the observations already stored in the case state. It then identifies the next unresolved requirement, selects a compatible tool, builds schema-valid arguments, executes the call, and updates the case state with the returned observation. Execution stops when the expected output fields are present or when the registry offers no action for the remaining requirement. Unresolved requirements are recorded explicitly in the case state.

Every observation is stored with its case, timepoint, source path, tool name, parameters, and execution status. Spatial outputs include key slices, bounding boxes, and masks; quantitative outputs include diameter, volume, attenuation, and longitudinal change; text outputs include findings, negation, and extracted measurements. Derived quantities retain links to their source mask and voxel spacing. We call this structured collection of observations evidence memory. It is the part of the case state on which later tool choices, measurements, and verification decisions depend.

Algorithm 1 ReAct-MCP inference procedure of CT-Agent

Input: Structured task object T, MCP tool registry R
Output: Task state S

$S \leftarrow \mathrm{Initialize}(T)$
while not $\mathrm{Completed}(S, T)$ **do**
  $g \leftarrow \mathrm{Reason}(T, S)$
  $m \leftarrow \mathrm{SelectTool}(g, R)$
  $x \leftarrow \mathrm{BuildInput}(g, S, \mathrm{Schema}(m))$
  $o \leftarrow \mathrm{Execute}(m, x)$
  $S \leftarrow \mathrm{UpdateState}(S, o)$
end while
return S

In Algorithm 1, T is the structured task object defined in Section 3.1, R is the MCP tool registry, and S is the persistent case state. $\mathrm{Reason}(T, S)$ produces the current goal g from the task and recorded

observations. $\mathrm{SelectTool}(g, R)$ selects a compatible tool m, and $\mathrm{BuildInput}$ constructs schema-valid arguments x. $\mathrm{Execute}(m, x)$ returns observation o, which $\mathrm{UpdateState}(S, o)$ writes back to the case state. $\mathrm{Completed}(S, T)$ checks whether the expected output fields in T are present in S.

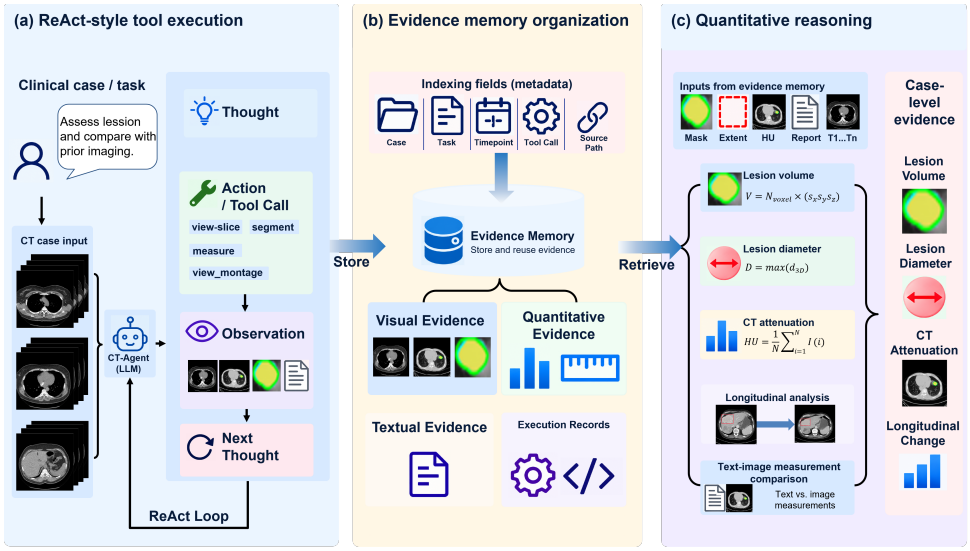


Figure 3: Evidence memory and quantitative reasoning. Visual, quantitative, textual, and execution observations remain linked to their case, source, and tool call.

Figure 3 presents evidence memory as a structured record of tool execution. A mask, measurement, report statement, and execution record are stored as different evidence types and remain linked by case, timepoint, and provenance. These links allow a derived measurement or longitudinal comparison to be traced back to the observation that produced it.

3.3 Verification and Output

After the required observations have been collected, the report module produces a candidate output in the requested format. Verification checks whether each finding or measurement has a corresponding record in evidence memory, whether outputs from invoked tools agree, and whether required fields, units, ranges, and uncertainty statements are present. Rule-based checks handle schema and numerical validity. An LLM judge (Zheng et al., 2023) receives only the candidate and the relevant evidence records when assessing semantic support.

A supported candidate is released with links to its source evidence. When verification fails, it returns a structured failure record that identifies an unsupported claim, a missing required field, or a cross-tool conflict. The controller can then obtain another observation, revise the candidate, or preserve the uncertainty for physician review. Automatic correction requires a guard defined on the separate calibration cases. This guard limits revisions based on additional tool results whose reliability varies across cases.

Figure 4 places verification inside the execution loop. Evidence-supported candidates leave the loop. Missing support or cross-tool conflict returns a structured reason to the controller, allowing the next action to target the specific evidence gap.

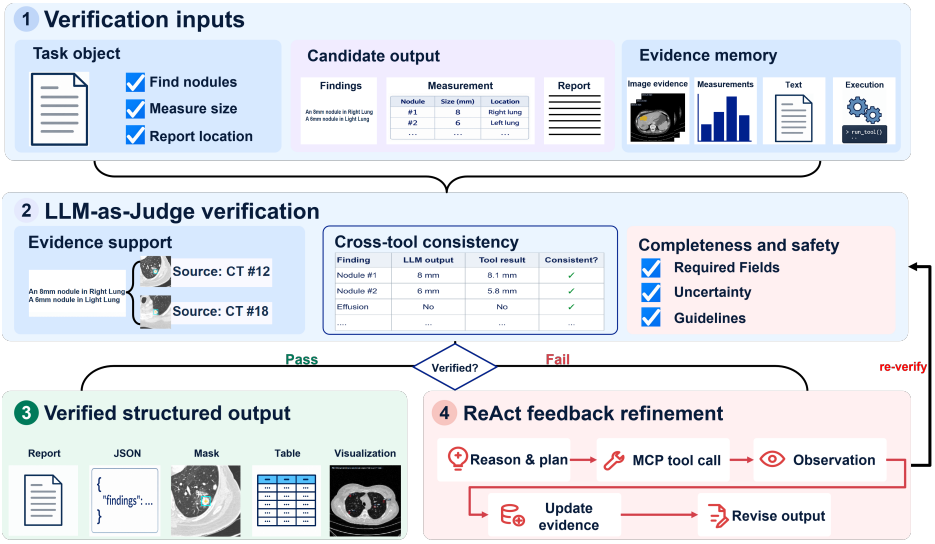


Figure 4: Evidence verification and feedback refinement. The verifier checks support, cross-tool consistency, completeness, and safety before accepting the result or returning a structured failure record to the execution loop.

4 Experiments and Results

4.1 Comparison with 3DMedAgent on 3D lesion segmentation

We compare CT-Agent with 3DMedAgent (Wang et al., 2026) on 3D lesion segmentation and quantitative measurement using the LiTS liver tumor dataset (Bilic et al., 2023). The experiment includes 22 cases with manual tumor masks, which are used as the reference standard. Evaluation metrics include the number of usable masks, Dice, IoU, voxel recall, center distance, maximum diameter error, and volume error.

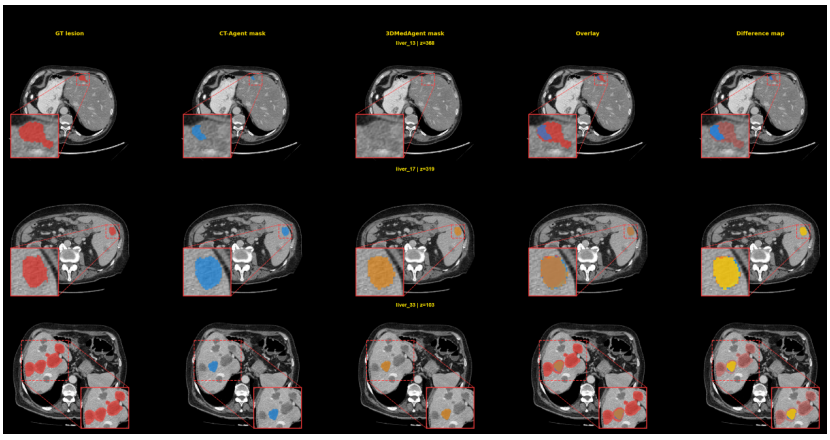


Figure 5: Liver tumor segmentation on LiTS. Rows are cases; columns show the reference, CT-Agent, 3DMedAgent, overlay, and difference map. Red denotes the reference, blue CT-Agent, orange 3DMedAgent, and yellow disagreement.

CT-Agent produces usable masks for all cases and improves Dice, IoU, center distance, maximum diameter error, and volume error over 3DMedAgent (Table 2). Voxel recall is similar, but CT-Agent gives more stable boundary overlap and quantitative measurements. Figure 5 shows representative cases.

Table 2: Quantitative comparison between CT-Agent and 3DMedAgent on LiTS liver tumor segmentation.

Metric	CT-Agent	3DMedAgent
Usable masks	22 / 22	22 / 22
Mean Dice	0.731	0.715
Mean IoU	0.593	0.575
Voxel recall	0.711	0.716
Center distance (mm)	10.999	12.604
Maximum diameter error (mm)	45.531	53.837
Volume error (%)	25.850	31.367

4.2 Multi-tool collaboration in airway segmentation

We evaluate multi-tool airway segmentation on 40 BronAtlas chest CT scans with manual annotations converted to binary masks (Huang et al., 2025). ATM22 (Zhang et al., 2023) and AeroPath (Støverud et al., 2024) are the single-tool baselines. CT-Agent performs case-level quality assessment, candidate fusion, and precision control over their outputs. We report Dice, IoU, Precision, and Recall.

CT-Agent improves Dice, IoU, and Precision over both tools while maintaining similar Recall (Figure 6 and Table 3). ATM22 and AeroPath differ in distal branch breaks, missed regions, and false-positive extensions, providing complementary candidates for case-level selection and fusion.

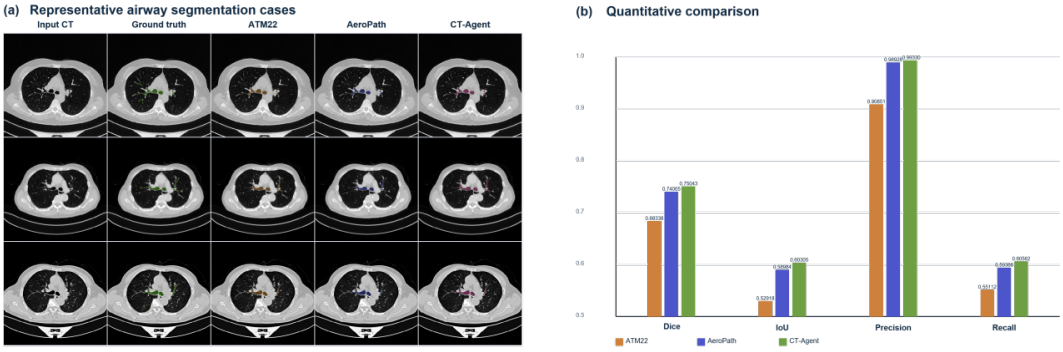


Figure 6: Airway segmentation on BronAtlas. (a) Three representative cases with input CT, reference, ATM22, AeroPath, and CT-Agent outputs. (b) Results over 40 cases.

Table 3: Quantitative results on BronAtlas airway segmentation.

Method	Dice	IoU	Precision	Recall
ATM22	0.68338	0.52918	0.90851	0.55112
AeroPath	0.74005	0.58984	0.98928	0.59366
CT-Agent	0.75043	0.60305	0.99330	0.60562

4.3 Task transfer and adaptation of CT-CLIP

We transfer CT-CLIP (Hamamci et al., 2026), originally developed for chest CT, to abnormality screening of the gallbladder/biliary system, pancreas, and spleen. Direct transfer matches organ-specific prompts to global CT representations and applies a fixed threshold of 0.5.

The public weights and image regions are closer to chest tasks, whereas abdominal findings are often confined to a local organ region. Global representations can therefore dilute the target signal. Moreover, CT-CLIP similarity scores support ranking but are not calibrated abnormality probabilities. The fixed threshold consequently produces zero sensitivity for all three abdominal tasks.

CT-Agent replaces the global input with an organ ROI, converts CT-CLIP outputs into task features for a lightweight ExtraTrees adapter (Geurts et al., 2006), and calibrates a separate threshold for each organ. These changes adapt both the representation and decision rule to the target task.

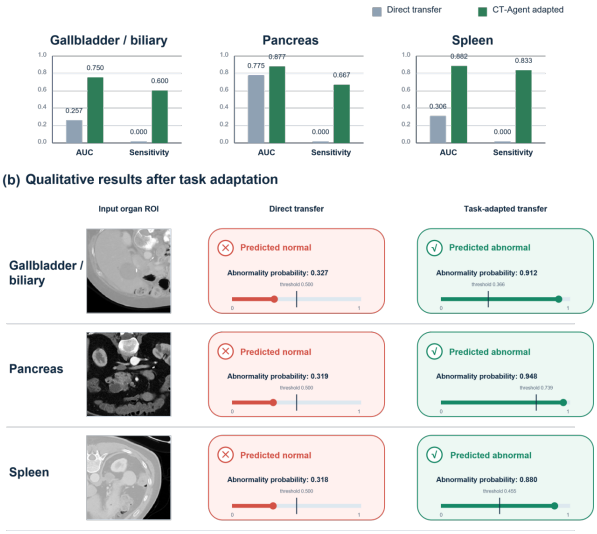


Figure 7: CT-CLIP transfer to abdominal screening: (a) metrics; (b) ROI inputs and predictions.

ROI inputs, the task adapter, and organ-specific calibration improve AUC and sensitivity for all three organs and recover positive predictions (Figure 7 and Table 4). CT-CLIP can serve as a representation tool within CT-Agent, but cross-anatomy transfer still requires task-specific inputs and calibration. These thresholds were calibrated within the experimental split and require independent testing.

Table 4: Abdominal abnormality screening results with direct CT-CLIP transfer and CT-Agent adaptation.

Task	Direct AUC	Direct sensitivity	Adapted AUC	Adapted sensitivity	Adapted accuracy
Gallbladder/biliary abnormality	0.257	0.000	0.750	0.600	0.853
Pancreatic abnormality	0.775	0.000	0.877	0.667	0.833
Splenic abnormality	0.306	0.000	0.882	0.833	0.833

4.4 Evidence traceability and output verification

We evaluate evidence traceability as the ability to connect a candidate output to source-linked intermediate records and to use those records to audit the output. We test this property on pulmonary-nodule assertions from the 40 CT-RATE test cases used in the complete-workflow evaluation (Appendix A). Without evidence verification and feedback refinement, the system retains the assertion produced by CT-CHAT. With both modules enabled, it compares that assertion with the nodule-detection record stored in evidence memory. Agreement preserves the assertion, while a conflict triggers a correction check based on a separate 20-case calibration set. The positive auto-correction threshold is 0.9985; this score range contains at least three supporting calibration cases and no false positives. Conflicts outside the guard retain the original assertion and are flagged for review. Test labels and reference reports are unavailable to the controller, expert tools, and verification module.

The verifier finds 18 disagreements between CT-CHAT and the detector. Five satisfy the calibration guard and lead to a revised assertion: four revisions correct an initial error, while one introduces an error. Figure 8 reports the corresponding changes in classification metrics and case outcomes. Accuracy, sensitivity, and F1 score increase after verification and refinement, while specificity decreases.

These results show that the retained evidence is usable for case-level auditing and correction, but they do not measure how efficiently a radiologist can inspect the record.

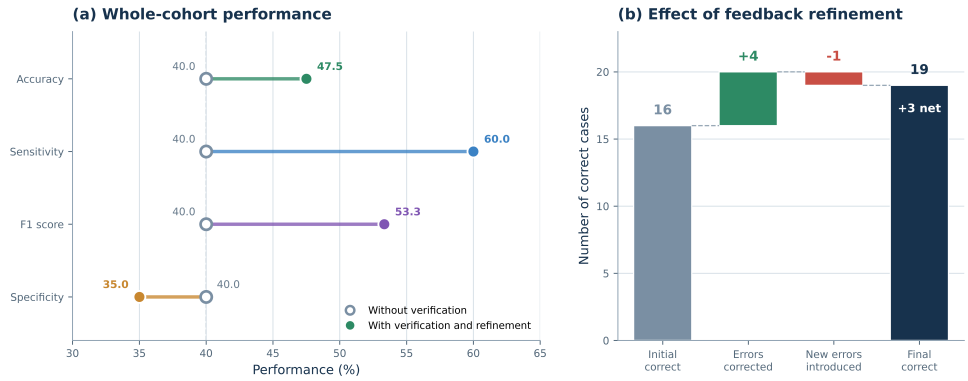


Figure 8: Effect of evidence verification and feedback refinement on pulmonary-nodule assertions. (a) Accuracy, sensitivity, and F1 score increase after verification and refinement, while specificity decreases. (b) Across 40 test cases, feedback refinement corrects four initial errors and introduces one new error, increasing the number of correct assertions from 16 to 19.

5 Discussion

The experiments separate sources of value that are often conflated in evaluations of medical agents. LiTS evaluates the complete segmentation and measurement path relative to another agent. BronAtlas evaluates coordination of existing candidates. The CT-CLIP experiment tests changes to inputs and decision rules during cross-anatomy transfer. CT-RATE tests whether retained evidence can expose cross-tool conflicts and support output verification. The supplementary CheXpert experiment isolates structured observation handling from the classifier itself. Together, these experiments characterize the contribution of the execution layer across complementary settings.

The results also support retaining intermediate evidence as structured data. A mask supports volume calculation when its geometry and source are preserved. A detector score can guide report revision when the disease target, threshold, and calibration provenance are known. CT-Agent uses these structured records to connect language-model planning and semantic comparison with image computation and numerical checks performed by dedicated tools and rules.

Several limitations constrain the present conclusions. LiTS includes 22 evaluated cases, and the CT-RATE test contains 40 cases from one public source. The CT-CLIP thresholds are calibrated within the experimental split. Open-source tools may differ from their originally reported settings. The current verification guard improves sensitivity and lowers specificity. Radiologist review therefore remains required. Robustness across institutions, scanners, disease prevalence, and prospective clinical workflows requires external evaluation.

6 Conclusion

CT-Agent treats heterogeneous CT analysis as execution over a persistent case state. It connects language-model planning with expert imaging tools, retains source-linked spatial and quantitative evidence, and checks candidate outputs against that evidence. Results across segmentation, tool coordination, cross-anatomy adaptation, and complete-workflow testing show measurable benefits and clear failure modes. Future work will validate tool selection, calibration, and verification on larger external cohorts with expert review.

AI USE STATEMENT

In this work, generative AI tools assisted with literature discovery, language editing and translation, LaTeX and reference formatting, figure layout, experimental-design feedback, software implementation, experiment orchestration, debugging, and interpretation of intermediate results. They were not used to generate experimental data or alter reported quantitative results. The authors independently checked the cited literature, code, data processing, calculations, figures, and reported results. The authors take responsibility for the final content of this work.

ETHICS STATEMENT

This study uses previously collected medical imaging datasets for research evaluation. The proposed system is intended as a research prototype for CT image analysis and should not be used for clinical decision making without prospective validation, expert review, and local regulatory approval.

REPRODUCIBILITY STATEMENT

The main paper describes the task settings, datasets, evaluation metrics, tool-level comparisons, adaptation procedure, and evidence-verification ablation. Appendix A reports the complete-workflow protocol, and Appendix B reports the supplementary chest X-ray experiment. Code, prompts, preprocessing scripts, tool configurations, and threshold calibration details will be released as supplementary material or through an anonymous repository.

REFERENCES

Anthropic. Introducing the model context protocol. Anthropic News, 2024. URL https://www.anthropic.com/news/model-context-protocol.

F. Bai, Y. Du, T. Huang, et al. M3D: advancing 3d medical image analysis with multi-modal large language models. *arXiv preprint arXiv:2404.00578*, 2024. doi: 10.48550/arXiv.2404.00578.

Patrick Bilic, Patrick Christ, Hongwei Bran Li, Eugene Vorontsov, Avi Ben-Cohen, Georgios Kaissis, Adi Szeskin, Colin Jacobs, Gabriel Efrain Humpire Mamani, Gabriel Chartrand, et al. The liver tumor segmentation benchmark (LiTS). *Medical Image Analysis*, 84:102680, 2023. doi: 10.1016/j.media.2022.102680.

M. Jorge Cardoso, Wenqi Li, Richard Brown, Nic Ma, Eric Kerfoot, Yiheng Wang, Benjamin Murrey, Andriy Myronenko, Can Zhao, Dong Yang, Vishwesh Nath, et al. MONAI: An open-source framework for deep learning in healthcare. *arXiv preprint arXiv:2211.02701*, 2022. doi: 10.48550/arXiv.2211.02701. URL https://arxiv.org/abs/2211.02701.

Wenting Chen, Yi Dong, Zhaojun Ding, Yucheng Shi, Yifan Zhou, Fang Zeng, Yijun Luo, Tianyu Lin, Yihang Su, Yichen Wu, Kai Zhang, Zhen Xiang, Tianming Liu, Ninghao Liu, Lichao Sun, Yixuan Yuan, and Xiang Li. RadFabric: Agentic AI system with reasoning capability for radiology. *arXiv preprint arXiv:2506.14142*, 2025. doi: 10.48550/arXiv.2506.14142. URL https://arxiv.org/abs/2506.14142.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026. doi: 10.48550/arXiv.2606.19348. URL https://arxiv.org/abs/2606.19348.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006. doi: 10.1007/s10994-006-6226-1.

I. E. Hamamci, S. Er, and B. Menze. CT2Rep: automated radiology report generation for 3d medical imaging. In *Medical Image Computing and Computer Assisted Intervention—MICCAI 2024*, pp. 476–486, Cham, 2024. Springer Nature Switzerland. doi: 10.1007/978-3-031-72390-2_45.

I. E. Hamamci, S. Er, C. Wang, et al. Generalist foundation models from a multimodal dataset for 3d computed tomography. *Nature Biomedical Engineering*, 10(8):1610–1628, 2026. doi: 10.1038/s41551-025-01599-y.

Yufan He, Pengfei Guo, Yucheng Tang, Andriy Myronenko, Vishwesh Nath, Ziyue Xu, Dong Yang, Can Zhao, Benjamin Simon, Mason Belue, Stephanie A. Harmon, Baris Turkbey, Daguang Xu, and Wenqi Li. VISTA3D: a unified segmentation foundation model for 3d medical imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20863–20873, 2025. doi: 10.1109/CVPR52734.2025.01943.

J. Hofmanninger, F. Prayer, J. Pan, et al. Automatic lung segmentation in routine imaging is primarily a data diversity problem, not a methodology problem. *European Radiology Experimental*, 4: 50, 2020. doi: 10.1186/s41747-020-00173-2.

Wenhao Huang, Haifan Gong, Huan Zhang, Yu Wang, Xiang Wan, Guanbin Li, Haofeng Li, and Hong Shen. BCNet: Bronchus classification via structure guided representation learning. *IEEE Transactions on Medical Imaging*, 44(1):489–498, 2025. doi: 10.1109/TMI.2024.3448468.

Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 590–597, 2019. doi: 10.1609/aaai.v33i01.3301590.

F. Isensee, P. F. Jaeger, S. A. A. Kohl, et al. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2):203–211, 2021. doi: 10.1038/s41592-020-01008-z.

C. Y. Li, K. J. Chang, C. F. Yang, et al. Towards a holistic framework for multimodal LLM in 3d brain CT radiology report generation. *Nature Communications*, 16:2258, 2025. doi: 10.1038/s41467-025-57426-0.

Yi Lin, Yihao Ding, Yonghui Wu, and Yifan Peng. MARCH: Multi-agent radiology clinical hierarchy for CT report generation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 273–285, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-short.24. URL https://aclanthology.org/2026.acl-short.24/.

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

Yuren Mao, Wenyi Xu, Yuyang Qin, and Yunjun Gao. CT-Agent: A multimodal-LLM agent for 3d CT radiology question answering. *arXiv preprint arXiv:2505.16229*, 2025. doi: 10.48550/arXiv.2505.16229. URL https://arxiv.org/abs/2505.16229.

Project MONAI. Lung nodule CT detection model bundle, version 0.6.10. MONAI Model Zoo, 2025. URL https://huggingface.co/MONAI/lung_nodule_ct_detection.

Tuoshi Qi, Shenshen Bu, Yingfei Xiang, and Zhiming Dai. EviAgent: Evidence-driven agent for radiology report generation. *arXiv preprint arXiv:2603.13956*, 2026. doi: 10.48550/arXiv.2603.13956. URL https://arxiv.org/abs/2603.13956.

Mélanie Roschewitz, Kenneth Styppa, Yitian Tao, Jiwoong Sohn, Jean-Benoit Delbrouck, Benjamin Gundersen, Nicolas Deperrois, Christian Bluethgen, Julia Vogt, Bjoern Menze, Farhad Nooralahzadeh, Michael Krauthammer, and Michael Moor. RadAgent: A tool-using AI agent for stepwise interpretation of chest computed tomography. *arXiv preprint arXiv:2604.15231*, 2026. doi: 10.48550/arXiv.2604.15231. URL https://arxiv.org/abs/2604.15231.

Karen-Helene Støverud, David Bouget, André Pedersen, Håkon Olav Leira, Tore Amundsen, Thomas Langø, and Erlend Fagertun Hofstad. AeroPath: an airway segmentation benchmark dataset with challenging pathology and baseline method. *PLOS ONE*, 19(10):e0311416, 2024. doi: 10.1371/journal.pone.0311416.

Ziyue Wang, Linghan Cai, Chang Han Low, Haofeng Liu, Junde Wu, Jingyu Wang, Rui Wang, Lei Song, Jiang Bian, Jingjing Fu, and Yueming Jin. 3DMedAgent: Unified perception-to-understanding for 3d medical analysis. *arXiv preprint arXiv:2602.18064*, 2026. doi: 10.48550/arXiv.2602.18064. URL https://arxiv.org/abs/2602.18064.

Jakob Wasserthal, Hanns-Christian Breit, Manfred T. Meyer, Maurice Pradella, Daniel Hinck, Alexander W. Sauter, Tobias Heye, Daniel Boll, Joshy Cyriac, Shan Yang, Michael Bach, and Martin Segeroth. Totalsegmentator: robust segmentation of 104 anatomic structures in CT images. *Radiology: Artificial Intelligence*, 5(5):e230024, 2023. doi: 10.1148/ryai.230024.

C. Wu, X. Zhang, Y. Zhang, et al. Towards generalist foundation model for radiology by leveraging web-scale 2d and 3d medical data. *Nature Communications*, 16:7866, 2025. doi: 10.1038/s41467-025-62385-7.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

M. Zhang, Y. Wu, H. Zhang, et al. Multi-site, multi-domain airway tree modeling. *Medical Image Analysis*, 90:102957, 2023. doi: 10.1016/j.media.2023.102957.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623, 2023. doi: 10.52202/075280-2020.

A COMPLETE-WORKFLOW EVALUATION

We evaluated case-level chest CT analysis on the CT-RATE validation set (Hamamci et al., 2026). Twenty cases were used for calibration, and a separate, balanced set of 40 cases was used for testing. Each case included a three-dimensional CT volume and a request for a complete chest CT analysis. Available clinical information was included in the structured task object. Pulmonary-nodule labels derived from the accompanying radiology reports were reserved for evaluation.

The DeepSeek-V4-Pro controller received the structured task object and the MCP tool registry and constructed an execution plan from the task requirements and tool descriptions. The resulting sequence called LungMask (Hofmanninger et al., 2020) for five-lobe segmentation, the MONAI lung-nodule detection bundle (Cardoso et al., 2022; Project MONAI, 2025) for pulmonary-nodule detection, AeroPath (Støverud et al., 2024) for airway segmentation, and CT-CHAT (Hamamci et al., 2026) for report generation.

Masks, bounding boxes, confidence scores, report text, call parameters, and execution status were written to evidence memory. Lung and airway volumes and lung attenuation were calculated from the segmentation masks and image geometry. After CT-CHAT produced a candidate report, the verification module compared its findings with the stored visual and quantitative evidence. CT-Agent then exported the structured record, masks, measurements, report, and visualization page.

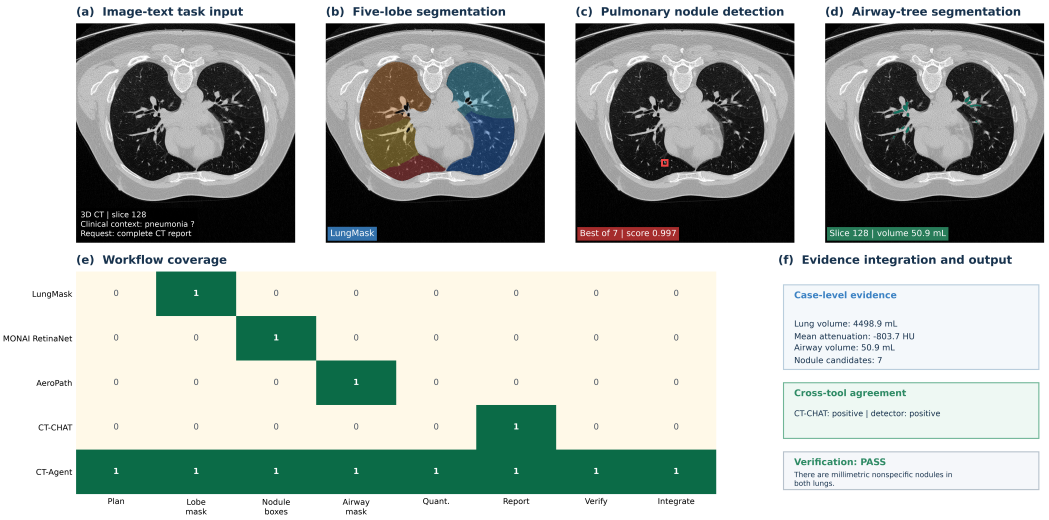


Figure A1: Representative result and capability coverage of the complete CT-Agent chest CT workflow. (a) A three-dimensional chest CT, clinical information, and a report-generation request form the image–text task input. (b–d) LungMask, MONAI 3D RetinaNet, and AeroPath produce a five-lobe mask, pulmonary-nodule candidates, and an airway mask, respectively; all displayed results correspond to axial slice 128. (e) Coverage of individual expert tools and CT-Agent across the workflow, where 1 denotes a supported stage and 0 an unsupported stage. (f) CT-Agent integrates imaging and quantitative evidence, checks agreement across tools, and returns a verified case-level output.

Figure A1 presents one representative case. The same CT volume was used to obtain the five-lobe mask, pulmonary-nodule candidates, airway mask, quantitative measurements, and candidate report. The lobe, nodule, and airway results are displayed on axial slice 128. Volumes and CT attenuation were calculated from the full three-dimensional outputs. CT-CHAT and the nodule detector both returned a positive finding for this case, and the verification result was pass.

All 40 test cases completed task planning, tool invocation, quantitative analysis, report generation, evidence verification, and result export. Every case yielded the required lobe mask, nodule candidates, airway mask, measurements, and candidate report, giving a workflow completion rate and

evidence completeness rate of 100%. The controller produced schema-compliant tool calls for every case.

As summarized in Figure A1(e), LungMask, MONAI 3D RetinaNet, AeroPath, and CT-CHAT cover lobe segmentation, nodule detection, airway segmentation, and report generation as individual expert tools. CT-Agent places these tools in a shared case workflow and adds task planning, quantitative analysis, evidence verification, and output integration. Its final output therefore contains imaging results, measurements, report text, and verification status within one case record.

B Supplementary Chest X-Ray Evaluation

We evaluate structured ReAct observation handling on 200 CheXpert cases with 14 labels (Irvin et al., 2019). CT-Agent-CXR records classifier probabilities as structured observations, whereas RadFabric-style reasoning (Chen et al., 2025) passes the same values directly to an LLM prompt. By holding the classifier inputs constant and changing the execution layer, this experiment tests the structured ReAct design and whether the agent framework can extend to chest radiography without changing its core orchestration mechanism.

Table B1: Results of CT-Agent-CXR and RadFabric-style reasoning on CheXpert.

Method	Cases	Macro AUC	Accuracy	Sensitivity	Specificity
CT-Agent-CXR	200	0.850	0.766	0.699	0.725
RadFabric-style reasoning	200	0.828	0.733	0.679	0.687

With identical classifier inputs, CT-Agent-CXR improves all four metrics over RadFabric-style reasoning in Table B1. The result supports the structured ReAct execution design in this setting and demonstrates that the agent framework can incorporate another imaging modality through task-specific tools and schemas.