# Learning to Be A Doctor: Searching for Effective Medical Agent Architectures

Yangyang Zhuang<sup>1</sup>, Wenjia Jiang<sup>1,2</sup>, Jiayu Zhang<sup>3,4</sup>, Ze Yang<sup>5</sup> Joey Tianyi Zhou<sup>6,7</sup> Chi Zhang<sup>1</sup>, 
<sup>1</sup>AGI Lab, Westlake University, <sup>2</sup>Henan University, <sup>3</sup>Affiliated Hospital of Xuzhou Medical University
<sup>4</sup>Xuzhou Medical University <sup>5</sup>Nanyang Technological University <sup>6</sup>IHPC, Agency for Science, Technology and Research, Singapore 
yyzhuang0211@gmail.com chizhang@westlake.edu.cn,

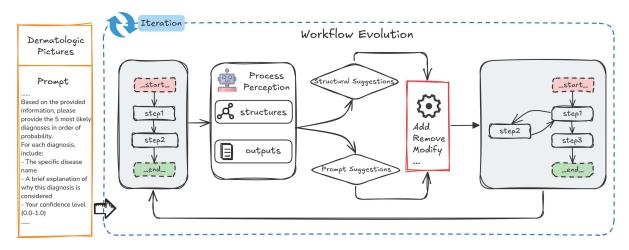


Figure 1: **Workflow Evolution Diagram.** This figure presents an iterative framework for workflow evolution in medical diagnosis tasks. Starting with an initial workflow, the system performs process perception to analyze its structure and outputs. Based on this analysis, both structural and prompt-level suggestions are generated to guide workflow refinement. Through repeated iterations, the workflow is progressively optimized to improve diagnostic accuracy and alignment with task objectives.

# **Abstract**

Large Language Model (LLM)-based agents have demonstrated strong capabilities across a wide range of tasks, and their application in the medical domain holds particular promise due to the demand for high generalizability and reliance on interdisciplinary knowledge. However, existing medical agent systems often rely on static, manually crafted workflows that lack the flexibility to accommodate diverse diagnostic requirements and adapt to emerging clinical scenarios. Motivated by the success of automated machine learning (AutoML), this paper introduces a novel framework for the automated design of medical agent architectures. Specifically, we define a hierarchical and expressive agent search space that enables dynamic workflow adaptation through structured modifications at the node, structural, and framework levels. Our framework conceptualizes medical agents as graph-based architectures composed of diverse, functional node types and supports iterative self-improvement guided by diagnostic feedback. Experimental results on skin disease diagnosis tasks demonstrate that

the proposed method effectively evolves workflow structures and significantly enhances diagnostic accuracy over time. This work represents the first fully automated framework for medical agent architecture design and offers a scalable, adaptable foundation for deploying intelligent agents in real-world clinical environments.

#### 1 Introduction

Large Language Models (LLMs) (Mann et al., 2020; Workshop et al., 2022; Achiam et al., 2023) have shown remarkable capabilities in solving a wide range of tasks (Park et al., 2023; Shinn et al., 2023b). Leveraging these models, LLM-based agents are increasingly transforming how tasks are executed across various domains, such as automation (Li et al., 2024b; Zhang et al., 2025a; Jiang et al., 2025) and biology (Li et al., 2024a, 2025). These agents are highly adaptive, capable of learning from prior experience, and continuously refining their behavior over time.

One particularly promising application of LLM-based agents (Kim et al., 2024; Jiang et al., 2025) is in the healthcare domain (Wang et al., 2024b; Li

et al., 2023). The medical field demands high accuracy, efficiency, and scalability, traits that align well with the strengths of intelligent agents. In this setting, LLM-based agents are beginning to play crucial roles beyond serving as assistive tools (Li et al., 2024a; Kim et al., 2024; Li et al., 2025). They are involved in essential processes such as diagnosis, medical data analysis, and treatment recommendation. By reducing the burden on healthcare professionals and improving clinical outcomes, these agents bring significant value to modern healthcare systems.

Among various multi-agent system designs (Chen et al., 2024c; Liang et al., 2024; Wang et al., 2024a), they offer a natural fit for complex medical workflows (Li et al., 2024a; Kim et al., 2024). Medical imaging and diagnosis tasks generally involve multiple interdisciplinary departments, making coordination and collaboration essential. In this context, multi-agent systems can efficiently assign duties to corresponding specialized agents while maintaining coherence in overall decision-making. They also help bridge communication gaps between departments and enable standardized procedures across institutions.

However, current multi-agent systems are constrained by how their workflows are designed. Existing systems typically rely on static architectures that are hand-crafted (Li et al., 2024a; Kim et al., 2024) by domain experts based on prior experience. Though effective in restricted circumstances, these fixed workflows often fail to adapt to complex real-world diagnostic scenarios. For example, when diagnostic techniques advance or new imaging modalities are introduced, it can take considerable time and effort to redesign existing workflows. Moreover, for each new task or application scenario, developers are required to create customized workflows from scratch, limiting scalability and slowing down deployment.

To address the aforementioned challenges, we draw inspiration from the field of Automated Machine Learning (AutoML) (Elshawi et al., 2019; Zhang et al., 2021; Chen et al., 2024a), particularly neural architecture search (NAS) (Wistuba et al., 2019), to explore automated architecture design for medical agents. NAS has demonstrated its effectiveness in autonomously discovering high-performing neural architectures across a variety of tasks. Analogously, we argue that the automated design of agent architectures can enhance the adaptability of multi-agent systems in dynamic clinical

environments.

To this end, we represent the agent system as a dynamic, graph-based workflow (Zhang et al., 2025b; Chen et al., 2024b) that evolves based on feedback from LLMs. To facilitate effective workflow evolution, we define a comprehensive search space comprising valid evolution operations. The operations are categorized into three levels: nodelevel, structural-level, and framework-level. Nodelevel operations include adding or removing nodes and modifying attributes within individual nodes. To enable more sophisticated reasoning, structurallevel operations introduce mechanisms such as conditional decision-making, iterative loops, and parallel execution, thereby enriching the logical workflow and enhancing diagnostic capabilities. At a higher abstraction level, framework-level operations incorporate established agent frameworks such as Chain of Thought (Wei et al., 2022), Reflexion (Shinn et al., 2023a), and Round Table (Chen et al., 2024c) to guide the search process, further improving the system's reasoning and coordination abilities.

After defining the evolution operations, we first establish an initial baseline workflow with only a single node to interact with the LLM and obtain the diagnostic results. As shown in Figure 1, when diagnostic errors occur, the agent system can identify the root causes and provide both structural and prompt suggestions for workflow evolution. As more cases are processed, the workflow undergoes iterative refinement, progressively incorporating new nodes, structural logic, and frameworklevel reasoning strategies. Over time, this results in a more accurate, robust, and efficient diagnostic system. This continuous evolution strategy equips the agent system with the capacity for selfimprovement to handle ever-changing real-world complexities.

To validate the effectiveness of our approach, we conduct comprehensive experiments on two medical diagnostic benchmarks (Naqvi, 2023; Daneshjou et al., 2023). The results demonstrate that our method can automatically generate more efficient and accurate agent architectures, significantly outperforming baselines. Our main contributions are as follows:

- We propose the first framework for fully automated design of medical multi-agent systems using LLMs.
- We introduce a novel hierarchical search space

tailored for dynamic agent workflow evolution.

 We develop a self-improving architecture search algorithm that enables agents to optimize their workflows through diagnostic feedback.

### 2 Related Work

# 2.1 LLM-Based Agents in the Medical Domain

In recent years, the reasoning capabilities of LLMs have improved significantly (Chen et al., 2025), enabling their broader application in the medical domain to enhance diagnostic accuracy and treatment effectiveness (Hager et al., 2024; Schmidgall et al., 2024; Singhal et al., 2023). For instance, Agent Hospital (Li et al., 2025) leverages scenario simulation to improve agents' medical knowledge and diagnostic accuracy, offering a virtual environment for training and evaluation. Building on similar principles, SkinGPT-4 (Zhou et al., 2024) applies vision-language models to refine dermatological diagnosis, combining visual and textual data for precise outcomes. Meanwhile, MMedAgent (Li et al., 2024a) integrates toolchains to enable multi-modal task execution, supporting robust performance across varied medical applications. More recent efforts such as MDAgents (Kim et al., 2024) explore multi-agent collaboration to improve efficiency in complex medical workflows. Despite their strengths in specific task performance, current systems pay limited attention to the dynamic workflows, which are essential for adapting to the complexity and variability of real-world medical environments.

# 2.2 Multi-Agent Collaboration and Evolution

Recent studies emphasize the growing importance of agent collaboration and self-improvement in multi-agent systems based on LLMs. Common architectures include peer-to-peer setups, centralized controllers, and hierarchical frameworks (Chen et al., 2023; Tran et al., 2025). Collaborative and self-improving multi-agent systems have shown promise in diverse application domains such as MetaGPT (Hong et al., 2024), LLM-Blender (Jiang et al., 2023) and Multi-Agent Debate (Liang et al., 2024). Beyond these domains, recent work focuses on enhancing adaptability in open-ended settings by exploring self-evolving mechanisms:

EvoMAC (Hu et al., 2024) introduces a self-evolving collaboration network that mimics the backpropagation mechanism in neural networks. AutoAgents (Chen et al., 2024b) supports automatic agent generation and flexible collaboration. AFLOW (Zhang et al., 2025b) uses graph search to optimize task execution paths.

# 3 Methodology

We propose a dynamic framework that leverages LLMs to optimize medical agent workflows. In our approach, workflows are represented as graph-based structures (Zhang et al., 2025b; Chen et al., 2024b), designed to include two node types: tool nodes and basic nodes. We first introduce the functions and attributes of different node types. We then formulate a structured and constrained search space, which defines the set of structural operations available for workflow evolution. Finally, we describe the workflow evolution process, which analyzes workflow shortcomings based on diagnostic errors and generates actionable suggestions for improvement.

### 3.1 Workflow Node Definition

Our medical agent workflow is formalized as a graph (Zhang et al., 2025b; Chen et al., 2024b), where nodes represent specialized agents responsible for distinct functional roles, and edges encode the execution order between them. Nodes are categorized into two types: basic nodes and tool nodes. We describe each node type in detail below.

**Basic node.** A basic node directly interacts with the LLM by processing system and human prompts to generate task-specific responses. It serves as the fundamental building block of medical agent workflows. As illustrated in Figure 2(a), each basic node is configured with the following key attributes:

- Node Name: The name of the basic node.
- **System Prompt**: Defines the role or identity of the agent within the node.
- **Human Prompt**: It specifies the instructions for the agent to carry out the designated task.
- **Description**: It provides a concise explanation of the node's function or purpose.

**Tool node.** A tool node can invoke predefined external tools. For example, image searching retrieves supplementary information that enhances

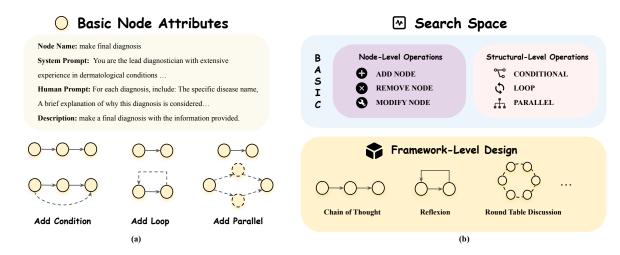


Figure 2: **Illustration of node attributes and hierarchical search space.** (a) Demonstrates the attributes of a basic node and how it can evolve into complex workflow structures. Each basic node contains configurable parameters. These attributes allow the agent to execute specific medical diagnostic tasks within the workflow. (b) Illustrates the proposed hierarchical search space for dynamic workflow design, consisting of node-level, structural-level, and framework-level operations. These enable fine-grained node manipulation, control flow structures, and emergent high-level reasoning patterns.

the agent's reasoning and decision-making. The tool node includes the following key attributes:

- Node Name: The name of the tool node.
- **Tool Name**: The name of the tool the node will use.
- **Description**: It provides a concise explanation of the node's function or purpose.

As a concrete example, we define an image search tool. This tool retrieves similar medical images from a reference database by leveraging the pre-trained CLIP (ViT-L/14) (Radford et al., 2021) model to extract and match image feature vectors without fine-tuning. Given a query image, it returns the top-k most similar cases with associated disease labels, which serve as diagnostic references for the LLM.

# 3.2 Search Space

To enable the discovery of better workflows, we propose a hierarchical search space that defines actionable strategies for agents across node-level, structural-level, and framework-level operations. As illustrated in Figure 2(b), this design allows for precise adjustments to both workflow structure and node prompts.

**Node-Level Operations.** Node-level operations involve modifying individual nodes to enhance both the functionality and efficiency of the workflow.

Specifically, we define the following three types of operations:

- Add: This operation adds a new node to the workflow, which can be either a basic node or a tool node. By adding such nodes, workflows can accommodate new functionalities and improve their reasoning and diagnostic performance when the complexity of tasks increases.
- Remove: It eliminates unnecessary or redundant nodes to improve workflow efficiency.
   Removing nodes helps streamline execution paths, resulting in a more concise and targeted process.
- **Modify:** This operation can update the prompt configurations of existing nodes, making the prompts more precise and better aligned with task requirements.

**Structural-Level Operations.** Structural operations enable modifications to the workflow's execution structure, including conditional branches, iterative loops, and parallel execution paths. These modifications enable the medical agent workflow to evolve from a simple linear sequence into a more adaptive and structured framework that more effectively supports complex diagnostic tasks. We define three types of structural operations.

- Conditional Structures: Conditional structures allow the workflow to take different execution paths based on logical conditions. This supports adaptive behavior by allowing the agent to select appropriate actions according to context, such as the patient symptoms or diagnostic outcomes.
- Loop Structures: Loop structures enable repeated execution of a node or sub-workflow until a termination condition is satisfied, such as achieving stable outputs or reaching a maximum iteration count. Additionally, loop structures support error correction and progressive refinement by allowing the medical workflow to revisit and improve prior decisions based on feedback.
- Parallel Structures: Parallel structures enable multiple nodes to conduct independent analyses of the same case simultaneously, each focusing on different clinical aspects such as image symptom description and differential diagnoses. By integrating these diverse perspectives through consensus or fusion mechanisms, the workflow achieves a more comprehensive and reliable diagnostic outcome.

Framework-Level Design. At the framework-level, the search space incorporates existing collaborative frameworks methods, including Chain of Thought (Wei et al., 2022), Reflexion (Shinn et al., 2023a), Round Table (Chen et al., 2024c), and Conquer-and-Merge Discussion (CMD) Framework (Wang et al., 2024a). Rather than directly adopting these structures, our approach dynamically composes customized collaboration patterns through combinations of basic nodes and structural operations.

To illustrate this process, we take the Round Table framework (Chen et al., 2024c) as an example. This framework organizes multiple expert nodes into a structured discussion process, where each node represents a specialized role (e.g., dermatologist, pathologist). In the first round, all expert nodes generate initial diagnostic suggestions. In subsequent rounds, the experts take turns refining their opinions in a sequential manner, using the previous responses and reasoning from other experts as context. This enables our framework to dynamically construct workflow structures that are functionally similar to established multi-agent

frameworks, while allowing these structures to be customized at both the node and structural levels to meet the specific requirements of the current diagnostic task.

### 3.3 Workflow Evolution

Our proposed workflow evolution process is an iterative refinement loop in which the LLM analyzes diagnostic failures from previous executions, identifies their root causes, and classifies them into two distinct error types: Image Understanding Errors and Diagnostic Errors. This classification guides targeted improvements to both the workflow structure and node prompts. The causes of these errors are collected and analyzed to generate improvement suggestions. These suggestions are then transformed into actionable workflow modifications. The modified workflow is validated and, if successful, deployed for the next diagnostic iteration. Through iterative cycles of diagnosis, analysis, feedback, and refinement, the workflow is continually improved and optimized. To better understand this evolution process, we provide a detailed description of the entire mechanism in the following.

First, LLM focuses on diagnosing workflow errors, and it aggregates outputs from all workflow nodes and classifies errors into two primary categories. **Image Understanding Errors**: Issues arising from inaccuracies in image descriptions, which compromise the accuracy of diagnostic decisions. **Diagnostic Errors**: Mistakes emerging from flawed reasoning or incorrect decisionmaking, even when the medical image descriptions are correct. Then, LLM traces the root causes of these errors, determining whether they originate from inaccurate image descriptions or errors in diagnostic inference.

Next, we need to analyze the structure of the workflow. Textual descriptions are commonly used to record workflow structures. However, they often fail to clearly represent the workflow's execution logic in a way that is easily interpretable by LLMs. In particular, the recorded node and edge relationships are not organized according to the workflow's execution sequence, but instead appended incrementally as structural updates are made. This can make it difficult for LLMs to accurately interpret the workflow structure, potentially leading to incorrect analysis of the execution logic.

To address this limitation, we convert the recorded structural information into Mer-

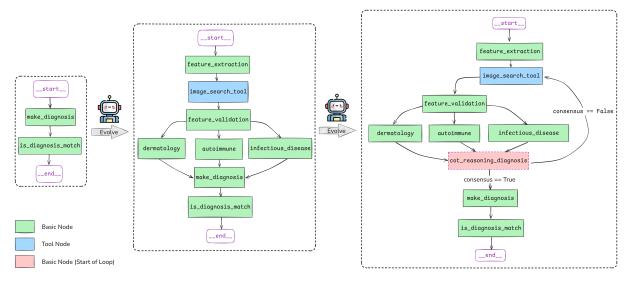


Figure 3: **Workflow evolution over iterations.** We visualize the diagnostic workflow at three stages: initial (Iteration 0), intermediate (Iteration 3), and final (Iteration 6). The evolution shows incremental improvements, including adding conditional branches, parallel paths, enhancing diagnostic accuracy and robustness through iterative refinement.

maid (Knut Sveidqvist, 2024) code and generate a corresponding visual workflow diagram. This visualization provides a clearer and more structured view of the workflow, enabling the LLM to better understand its execution logic and offer more accurate suggestions for improvement.

Having analyzed the workflow structure and the issues identified by the LLM, we now explain how the LLM translates these insights into actionable modifications for iterative refinement. Our framework implements a systematic refinement process to ensure safe and effective workflow evolution. First, generated suggestions are filtered to discard unrealistic actions, such as those that require external inputs beyond the agent's capabilities, retaining only feasible and actionable ones. These filtered suggestions are then categorized into two types: Structural suggestions, which aim to optimize the workflow's architecture (e.g., adding loops or parallel branches), and **Prompt suggestions**, which refine task-specific configurations such as node prompts to improve reasoning quality.

To enable automated processing, all suggestions are reformulated into standardized formats that clearly specify the intended modifications. Before processing, we define a set of prompt templates that outline the required parameters for the different types of node modifications. Based on these templates, the LLM reformulates each suggestion into a structured format, which provides the structured data required for updating the workflow.

Structural suggestions are carefully validated before integration, as they may break the normal execution of the workflow. To prevent such modifications, the framework incorporates a set of structural validation mechanisms to ensure that all modifications maintain the integrity of the workflow graph and its execution logic. To prevent duplicate edges, the workflow rejects any modification that attempts to add a new edge between two nodes that are already connected. This prevents redundant or conflicting execution paths. Additionally, the framework also limits the maximum number of iterations to prevent infinite loops and validates the directionality of edges when introducing cycles, ensuring that loops have a well-defined exit condition. These validation mechanisms work together to safeguard the workflow from invalid or disruptive changes, ensuring that all structural modifications are both logical and executable.

Prompt suggestions only affect the internal configuration of individual nodes and do not alter the workflow structure. As a result, they can be applied directly without additional validation. These modifications typically include updates to the system prompt, which provides a more accurate definition of the node's role or identity, and the human prompt, which clarifies the specific task a node should complete.

The entire process, from diagnosis and error analysis to workflow updates, can be viewed as an evolutionary cycle. During each iteration, multiple

disease cases are processed, and the workflow is subsequently refined to produce an updated version. This cycle is repeated iteratively until the accuracy of the final workflow on the validation set converges, indicating the completion of the evolution process.

# 4 Experiments

## 4.1 Experimental Setup

**Datasets.** To evaluate the effectiveness of our proposed framework, we conduct experiments on two dermatological image classification datasets (Daneshjou et al., 2023; Naqvi, 2023) to assess workflow evolution. Dermatology presents a challenging and representative domain for evaluating multi-agent architectures, as accurate diagnosis typically demands both specialized visual analysis and integrative reasoning. Below is a detailed description of the two datasets.

**Skin Concepts** (Daneshjou et al., 2023) extends the Fitzpatrick 17k skin disease dataset (Groh et al., 2022, 2021) and includes 3,230 images annotated with 48 clinically relevant skin concepts. For evaluation, we adopt a split of 50 images for training, 50 for validation, and 224 for testing, ensuring two test images per disease type to maintain balanced coverage across all categories.

Augmented Skin Conditions (Naqvi, 2023) consists of 2,394 synthetically augmented images across six dermatological conditions, with 399 images per class. For evaluation, we adopt a similar split of 50 training, 50 validation, and 120 test images, with 20 images per disease type, to ensure sufficient and balanced evaluation of diagnostic performance.

During data processing, we observed that the LLM rejected a small number of images depicting sensitive body areas due to content moderation policies. Additionally, some image data with broken URLs could not be downloaded. To ensure experimental integrity and enable fair and consistent evaluation across all agent architectures, we excluded these samples from our experiments. In total, 100 images from the **Skin Concepts** (Daneshjou et al., 2023) dataset were removed.

**Baselines.** To establish a comparative framework for evaluating the effectiveness of our workflow evolution methodology, we implemented three baseline approaches. The first baseline utilizes LLM to give a direct diagnosis of input images, without any intermediate reasoning or workflow ad-

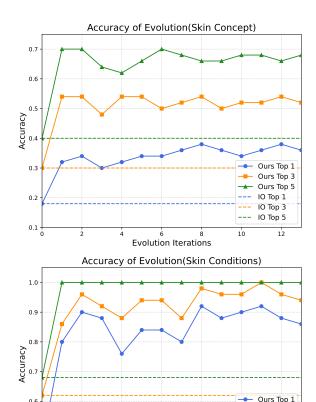


Figure 4: Top-k accuracy improves with workflow iterations, and convergence signals system stabilization using GPT-4o.

**Evolution Iterations** 

Ours Top 3 Ours Top 5

IO Top 1 IO Top 3

IO Top 5

10

justments. This provides a straightforward benchmark for assessing diagnostic accuracy. The second baseline augments the LLM with a CoT(Wei et al., 2022) prompting mechanism. This approach guides LLM to reason through intermediate steps before producing the final diagnosis. The third baseline adopts the Round Table (Chen et al., 2024c) framework, a multi-agent collaboration paradigm inspired by human expert panels. In this setup, multiple agents take turns to refine a shared solution through iterative deliberation. Each agent reviews the reasoning history and outputs of previous agents, then contributes an updated response based on its own perspective. This process continues for a fixed number of iterations or until convergence, aiming to achieve a more comprehensive and accurate final decision.

**Implementation Details.** Our experimental setup includes training, validation, and testing phases. During training, the workflow evolves through iterative refinement based on diagnostic outcomes.

Each updated workflow is validated after every batch to assess performance improvements. Once training completes, the final workflow is evaluated on the test set to measure overall performance.

Our framework extends LangGraph (Inc., 2023) with customized node designs to support flexible and modular workflow construction. To facilitate extensibility, we introduce a configurable tool list that allows users to easily add or integrate their own custom tools. As a built-in example, we include an image search tool designed to find visually similar images. We use the pre-trained CLIP (ViT-L/14) (Radford et al., 2021) model to extract feature vectors from images that are not included in the training, validation, or test sets. These vectors are then stored in the Pinecone (Pinecone Systems, 2023) vector database to support image similarity searches. When a new image is input, its feature vector is computed and compared against the stored vectors to enable fast and efficient similarity searches.

To evaluate the generality and robustness of our framework across diverse LLMs, we conduct experiments using the following models: GPT-4o-2024-05-13 (Achiam et al., 2023), Claude-3-5-sonnet-20240620 (Anthropic., 2024), and GPT-4o-mini-2024-07-18 (Menick et al., 2024). The temperature parameter is set to 1.0 to encourage diverse reasoning paths, and the random seed is 42.

**Evaluation Metrics.** To evaluate diagnostic models, we used Top-1, Top-3, and Top-5 accuracy metrics, comparing predictions against ground truth labels. During validation, these accuracies were tracked across workflow evolution to observe improvements, while in testing, final accuracies were compared with baselines to highlight the benefits of adaptive workflows.

In addition to accuracy, we assess the consistency of predictions across repeated inputs. We adopt the majority voting consensus metric, denoted as **cons@64** (Guo et al., 2025), which reflects how often a method arrives at a consistent diagnosis across multiple inputs of the same class (n=64). This metric offers a proxy for model robustness and reliability under input variance and sampling noise.

These metrics allow us to quantify both the progression of accuracy during workflow evolution and the overall generalization ability of the proposed method when tested on unseen data, particularly in diverse disease categories.

#### 4.2 Results

**Evolution Visualization.** To better understand how our workflow evolves, we visualize the structure at three representative stages: the initial workflow (Iteration 0), an intermediate state (Iteration 3), and the final evolved workflow (Iteration 6). Figure 3 presents a comparative view of these stages. These visualizations reveal how the system progressively incorporates structural refinements, such as conditional branches, parallel reasoning paths, and error-specific loops.

Cost and Complexity Analysis. The results in Table 2 show that the framework evolves into more complex workflows by the end of training. The final workflows used in testing contain more nodes and branches than the training-phase averages, indicating an increase in both structural complexity and token consumption driven by the evolutionary process.

**Performance Over Iterations.** To quantitatively evaluate the effectiveness of our workflow evolution framework, we track the diagnostic performance, measured by Top-1 accuracy, Top-3 accuracy, and Top-5 accuracy across multiple evolution iterations. At each iteration, the system refines its workflow based on prior diagnostic errors and structural improvements, allowing it to incrementally enhance its reasoning capabilities.

Figure 4 illustrates the iterative enhancement in accuracy as the workflow progresses, compared to the baseline accuracy achieved by directly inputting images into the LLM for diagnosis (IO line). The evolved workflow consistently surpasses the baseline performance, exhibiting an initial upward trend attributed to structural refinement and prompt optimization. This is followed by convergence as the system stabilizes into an effective configuration. This pattern underscores the framework's capacity to learn from experience and self-optimize over time. In the context of the Skin Concepts dataset (Daneshjou et al., 2023), the evolved workflow demonstrates robust generalization capabilities, achieving commendable results across all categories in the testing set.

**Evolution Results Analysis.** To comprehensively evaluate the effectiveness of our method in diagnostic tasks, we compare it against various baselines as shown in Table 1. Specifically, we report the Top-k diagnostic accuracy on dermatological tasks using the Skin Concepts (Daneshjou et al., 2023) and Skin Conditions (Naqvi, 2023) datasets across

Table 1: Top-k diagnostic accuracy (%) of different methods using GPT-40, GPT-40-mini and Claude 3.5 Sonnet on
Skin Concepts and Skin Conditions.

LLM	Method	Skin Concepts Accuracy (%)			Skin Conditions Accuracy (%)		
LLM	Method	Top-1	Top-3	Top-5	Top-1	Top-3	Top-5
	IO	20.27	30.63	36.04	50.83	78.33	86.67
CDT 4a	CoT (Wei et al., 2022)	18.47	28.83	33.78	55.83	76.67	82.50
GPT-40	Round Table (Chen et al., 2024c)	21.17	27.93	32.43	45.83	75.83	80.83
	Ours	29.28	40.09	50.45	90.83	95.00	100.00
	IO	11.71	20.72	23.87	27.50	69.17	80.83
GPT-40-mini	CoT (Wei et al., 2022)	6.31	15.32	24.32	22.50	65.00	84.17
GP 1-40-IIIIII	Round Table (Chen et al., 2024c)	10.81	19.82	23.42	25.83	70.00	78.33
	Ours	13.51	21.62	24.77	45.83	74.17	85.83
	IO	17.12	24.77	26.13	40.00	68.33	75.83
Claude 3.5 Sonnet	CoT (Wei et al., 2022)	14.86	22.07	24.32	36.67	63.33	73.33
	Round Table (Chen et al., 2024c)	15.77	22.52	26.58	35.00	66.67	74.17
	Ours	28.83	35.14	38.29	95.83	98.33	99.17

Table 2: Resource Consumption and Workflow Complexity. This table summarizes resource consumption and workflow complexity across different metrics during training (Evo) and testing phases.

Metric	Train (Evo)	Test
Number of Nodes	8.7	11
Number of Branches	4.0	5
Time (s)	210	621
Total Tokens (k)	361	349

three different LLMs.

Our approach consistently outperforms existing baselines, including IO, CoT (Wei et al., 2022), and Round Table (Chen et al., 2024c), across all LLMs and evaluation metrics. Our method achieves superior results in Top-1, Top-3, and Top-5 accuracies for both datasets. These results highlight the adaptability and robustness of our method across different LLMs and diverse medical diagnostic scenarios. The consistent superiority in Top-k accuracies underscores the effectiveness and reliability of our evolved workflow, making it a promising solution for improving diagnostic performance.

Workflow Stability. In addition to diagnostic accuracy, we assess the stability of our evolved workflows by evaluating their consensus behavior across multiple independent samples. We report per-class consensus accuracy (cons@64 (Guo et al., 2025)) for each method using the GPT-40 model on the Skin Concepts dataset (Daneshjou et al., 2023), as shown in Figure 5. Our method consistently achieves higher consensus scores across a wide range of conditions.

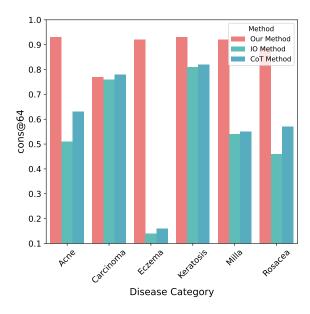


Figure 5: Per-disease consensus accuracy (cons@64) across different methods. This figure shows prediction stability across six disease categories, evaluated via majority-vote consensus over 64 samples per class. Our method achieves higher cons@64 scores, indicating greater diagnostic robustness and consistency.

# 4.3 Ablation Analysis

We conduct a comprehensive ablation study to evaluate the contribution of individual workflow modification actions in our evolution framework. Actions are classified into three categories: *Add*, *Modify*, and *Remove*. In the study, we assess the impact of three key operations: adding tool nodes, modifying node prompts, and removing nodes. By excluding each operation from the search space, we observe its contribution to overall performance. Table 3 shows that adding tool nodes and mod-

ifying prompts notably improve accuracy, while removing nodes has minimal impact on accuracy but enhances execution efficiency by reducing redundancy.

Table 3: Ablation results showing Top-k accuracies when specific operations are disabled. Parentheses indicate accuracy drops relative to the full model.

Operation	Top-1 Acc. (%)	Top-3 Acc. (%)	Top-5 Acc. (%)
Add Tool Node	21.62 (-7.66)	30.18 (-9.91)	36.94 (-13.51)
Modify Node Prompt	19.37 (-9.91)	27.93 (-12.16)	33.78 (-16.67)
Remove Node	28.83 (-0.45)	41.44 (+1.35)	50.90 (-0.45)

We further evaluate the framework under three different integration levels: node-level (N), node + structural-level (N+S), and full-level (N+S+F). The results show that incorporating structural-level modifications leads to performance improvement, with the full-level framework achieving the best results. This confirms the effectiveness of hierarchical integration in supporting comprehensive workflow evolution.

Table 4: Ablation results for integrating different operation levels (N, N+S, N+S+F). The fully-leveled framework delivers the best performance.

Setting	<b>Top-1</b> Acc. (%)	<b>Top-3</b> Acc. (%)	<b>Top-5</b> Acc. (%)
N	26.13	36.94	45.05
N+S	27.03	38.74	46.40
N+S+F	29.28	40.09	50.45

Together, these experiments provide a comprehensive understanding of how individual operations and integration levels contribute to performance, offering interpretable evidence for the design in our framework.

#### 5 Conclusion

Current medical diagnostic systems are often constrained by rigid, manually designed workflows that lack the flexibility to adapt to evolving clinical demands and diverse diagnostic requirements. To address this limitation, we propose a workflow evolution framework that enables the autonomous exploration and iterative refinement of agent-based diagnostic workflows. To enable a more responsive and robust diagnostic workflow, our framework introduces dynamic adaptability at both the structural and prompt levels, allowing it to evolve in response to diagnostic errors. In summary, our framework provides a scalable and adaptive solution for advancing multi-agent systems in diagnostic tasks.

It not only bridges the gap between static, expertdefined architectures and dynamic, self-improving methods, but also lays the groundwork for future adaptive diagnostic systems.

#### References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

Anthropic. 2024. Claude 3.5 sonnet.

Bo Chen, Xiangyu Zhao, Yejing Wang, Wenqi Fan, Huifeng Guo, and Ruiming Tang. 2024a. A comprehensive survey on automated machine learning for recommendations. *ACM Transactions on Recommender Systems*, 2(2):1–38.

Guangyao Chen, Siwei Dong, Yu Shu, Ge Zhang, Jaward Sesay, Börje F. Karlsson, Jie Fu, and Yemin Shi. 2024b. Autoagents: A framework for automatic agent generation. *Preprint*, arXiv:2309.17288.

Justin Chen, Swarnadeep Saha, and Mohit Bansal. 2024c. ReConcile: Round-table conference improves reasoning via consensus among diverse LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7066–7085, Bangkok, Thailand. Association for Computational Linguistics.

Shuaihang Chen, Yuanxing Liu, Wei Han, Weinan Zhang, and Ting Liu. 2025. A survey on llm-based multi-agent system: Recent advances and new frontiers in application. *Preprint*, arXiv:2412.17481.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2023. Agentverse: Facilitating multiagent collaboration and exploring emergent behaviors. *Preprint*, arXiv:2308.10848.

Roxana Daneshjou, Mert Yuksekgonul, Zhuo Ran Cai, Roberto Novoa, and James Zou. 2023. Skincon: A skin disease dataset densely annotated by domain experts for fine-grained model debugging and analysis. *Preprint*, arXiv:2302.00785.

Radwa Elshawi, Mohamed Maher, and Sherif Sakr. 2019. Automated machine learning: State-of-the-art and open challenges. *arXiv preprint arXiv:1906.02287*.

Matthew Groh, Caleb Harris, Roxana Daneshjou, Omar Badri, and Arash Koochek. 2022. Towards transparency in dermatology image datasets with skin tone annotations by experts, crowds, and an algorithm. *Proceedings of the ACM on Human-Computer Interaction*, 6(CSCW2):1–26.

- Matthew Groh, Caleb Harris, Luis Soenksen, Felix Lau, Rachel Han, Aerin Kim, Arash Koochek, and Omar Badri. 2021. Evaluating deep neural networks trained on clinical images in dermatology with the fitzpatrick 17k dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1820–1828.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Paul Hager, Friederike Jungmann, Robbie Holland, Kunal Bhagat, Inga Hubrecht, Manuel M Knauer, Jakob Vielhauer, Marcus Makowski, Rickmer F Braren, Georgios Kaissis, and Daniel Rueckert. 2024. Evaluation and mitigation of the limitations of large language models in clinical decision-making. *Nature Medicine*, 30:2613 2622.
- Sirui Hong, Mingchen Zhuge, Jiaqi Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. Metagpt: Meta programming for a multi-agent collaborative framework. *Preprint*, arXiv:2308.00352.
- Yue Hu, Yuzhu Cai, Yaxin Du, Xinyu Zhu, Xiangrui Liu, Zijie Yu, Yuchen Hou, Shuo Tang, and Siheng Chen. 2024. Self-evolving multi-agent collaboration networks for software development. *Preprint*, arXiv:2410.16946.
- LangChain Inc. 2023. Langgraph.
- Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *Preprint*, arXiv:2306.02561.
- Wenjia Jiang, Yangyang Zhuang, Chenxi Song, Xu Yang, Joey Tianyi Zhou, and Chi Zhang. 2025. Appagentx: Evolving gui agents as proficient smartphone users. *arXiv preprint arXiv:2503.02268*.
- Yubin Kim, Chanwoo Park, Hyewon Jeong, Yik Siu Chan, Xuhai Xu, Daniel McDuff, Hyeonhoon Lee, Marzyeh Ghassemi, Cynthia Breazeal, and Hae Won Park. 2024. Mdagents: An adaptive collaboration of llms for medical decision-making. *Preprint*, arXiv:2404.15155.
- Knut Sveidqvist. 2024. Mermaid documentation. https://mermaid.js.org/. Accessed: 2025-07-30.
- Binxu Li, Tiankai Yan, Yuanting Pan, Jie Luo, Ruiyang Ji, Jiayuan Ding, Zhe Xu, Shilong Liu, Haoyu Dong, Zihao Lin, and Yixin Wang. 2024a. MMedAgent: Learning to use medical tools with multi-modal agent. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8745–8760, Miami, Florida, USA. Association for Computational Linguistics.

- Junkai Li, Yunghwei Lai, Weitao Li, Jingyi Ren, Meng Zhang, Xinhui Kang, Siyu Wang, Peng Li, Ya-Qin Zhang, Weizhi Ma, and Yang Liu. 2025. Agent hospital: A simulacrum of hospital with evolvable medical agents. *Preprint*, arXiv:2405.02957.
- Yanda Li, Chi Zhang, Wanqi Yang, Bin Fu, Pei Cheng, Xin Chen, Ling Chen, and Yunchao Wei. 2024b. Appagent v2: Advanced agent for flexible mobile interactions. *arXiv preprint arXiv:2408.11824*.
- Yunxiang Li, Zihan Li, Kai Zhang, Ruilong Dan, Steve Jiang, and You Zhang. 2023. Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge. *Cureus*, 15(6).
- Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17889–17904, Miami, Florida, USA. Association for Computational Linguistics.
- Ben Mann, Nick Ryder, Melanie Subbiah, J Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry, A Askell, S Agarwal, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 1(3):3.
- Jacob Menick, Kevin Lu, Shengjia Zhao, E Wallace,
   H Ren, H Hu, N Stathas, and F Petroski Such. 2024.
   Gpt-40 mini: advancing cost-efficient intelligence.
   Open AI: San Francisco, CA, USA.
- Syed Ali Raza Naqvi. 2023. Augmented skin conditions image dataset. Kaggle.
- Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.
- Inc. Pinecone Systems. 2023. Pinecone: The vector database for ai applications. A vector database designed for scalable and performant AI applications, enabling semantic search, metadata filtering, and real-time updates.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *Preprint*, arXiv:2103.00020.
- Samuel Schmidgall, Rojin Ziaei, Carl Harris, Eduardo Reis, Jeffrey Jopling, and Michael Moor. 2024. Agentclinic: a multimodal agent benchmark to evaluate ai in simulated clinical environments. *ArXiv*, abs/2405.07960.

- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023a. Reflexion: language agents with verbal reinforcement learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.
- Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023b. Reflexion: Language agents with verbal reinforcement learning, 2023. *URL https://arxiv.org/abs/2303.11366*, 1.
- Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Abubakr Babiker, Nathanael Schaerli, Aakanksha Chowdhery, Philip Mansfield, Dina Demner-Fushman, Blaise Aguera y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. 2023. Large language models encode clinical knowledge. *NATURE*, 620(7972):172+.
- Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D. Nguyen. 2025. Multi-agent collaboration mechanisms: A survey of llms. *Preprint*, arXiv:2501.06322.
- Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. 2024a. Rethinking the bounds of llm reasoning: Are multi-agent discussions the key? *arXiv preprint arXiv:2402.18272*.
- Sheng Wang, Zihao Zhao, Xi Ouyang, Tianming Liu, Qian Wang, and Dinggang Shen. 2024b. Interactive computer-aided diagnosis on medical image using large language models. *Communications Engineering*, 3(1):133.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.
- Martin Wistuba, Ambrish Rawat, and Tejaswini Pedapati. 2019. A survey on neural architecture search. *arXiv preprint arXiv:1905.01392*.
- BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. arXiv preprint arXiv:2211.05100.
- Chi Zhang, Henghui Ding, Guosheng Lin, Ruibo Li, Changhu Wang, and Chunhua Shen. 2021. Meta

- navigator: Search for a good adaptation policy for few-shot learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9435–9444.
- Chi Zhang, Zhao Yang, Jiaxuan Liu, Yanda Li, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. 2025a. Appagent: Multimodal agents as smartphone users. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–20.
- Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. 2025b. Aflow: Automating agentic workflow generation. *Preprint*, arXiv:2410.10762.
- Juexiao Zhou, Xiaonan He, Liyuan Sun, Jiannan Xu, Xiuying Chen, Yuetan Chu, Longxi Zhou, Xingyu Liao, Bin Zhang, Shawn Afvari, and Xin Gao. 2024. Pretrained multimodal large language model enhances dermatological diagnosis using skingpt-4. NATURE COMMUNICATIONS, 15(1).