

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 FEDAGENTBENCH: TOWARDS AUTOMATING REAL- WORLD FEDERATED MEDICAL IMAGE ANALYSIS WITH SERVER-CLIENT LLM AGENTS

006
007
008
009
010
011
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
031
032
033
034
035
036
037
Anonymous authors

Paper under double-blind review

ABSTRACT

000
001
002
003
004
005
006
007
008
009
010
011
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
031
032
033
034
035
036
037
Federated learning (FL) allows collaborative model training across healthcare sites without sharing sensitive patient data. However, real-world FL deployment is often hindered by complex operational challenges that demand substantial human efforts in cross-client coordination and data engineering. This includes: (a) selecting appropriate clients (hospitals), (b) coordinating between the central server and clients, (c) client-level data pre-processing, (d) harmonizing non-standardized data and labels across clients, and (e) selecting FL algorithms based on user instructions and cross-client data characteristics. However, the existing FL works overlook these practical orchestration challenges. These operational bottlenecks motivate the need for autonomous, agent-driven FL systems, where intelligent agents at each hospital client and the central server agent collaboratively manage FL setup and model training with minimal human intervention. To this end, we first introduce: (i) an agent-driven FL framework that captures key phases of real-world FL workflows from client selection to training completion, and (ii) a benchmark dubbed FedAgentBench that evaluates the ability of LLM agents to autonomously coordinate healthcare FL. Our framework incorporates 40 FL algorithms, each tailored to address diverse task-specific requirements and cross-client characteristics. Furthermore, we introduce a diverse set of complex tasks across 201 carefully curated datasets, simulating 6 modality-specific real-world healthcare environments, *viz.*, Dermatoscopy, Ultrasound, Fundus, Histopathology, MRI, and X-Ray. We assess the agentic performance of 14 open-source and 10 proprietary LLMs spanning small, medium, and large model scales. While some agent cores such as GPT-4.1 and DeepSeek V3 can automate various stages of the FL pipeline, our results reveal that more complex, interdependent tasks based on implicit goals remain challenging for even the strongest models.

1 INTRODUCTION AND BACKGROUND

000
001
002
003
004
005
006
007
008
009
010
011
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
031
032
033
034
035
036
037
Federated Learning (FL) (Li et al., 2021b; McMahan et al., 2017; Li et al., 2020a) allows collaborative model training across multiple healthcare institutions (*e.g.*, hospitals) without sharing raw medical data. A typical FL workflow involves several tightly coupled components: selecting suitable clients for training, preprocessing heterogeneous data locally, harmonizing labels and datasets across clients, coordinating client-server communication, selecting optimal FL algorithm, and aggregating model updates in the server. These components must be executed in a precise and orchestrated manner across multiple clients. Real-world execution of an FL pipeline necessitates close coordination by human data scientists and machine learning engineers in server and client locations, who are tasked with managing a range of demanding communicational and technical operations. These include selecting appropriate client nodes based on task relevance and resource availability, implementing local data preprocessing pipelines (*e.g.*, normalization, filtering, schema mapping), and harmonizing cross-site inconsistencies of data and label spaces. Additionally, they must determine the most suitable FL algorithms, and manage training schedules and aggregation strategies. This manual orchestration poses a significant barrier to scalable and real-time deployment of FL, particularly in sensitive domains like healthcare, where institutions store diverse yet complementary datasets that cannot be centralized due to privacy and compliance constraints. Moreover, many healthcare facilities, especially in low- and middle-income countries (LMICs) and rural areas, lack the resources to hire dedicated data scientists or machine learning engineers, further limiting their ability to participate

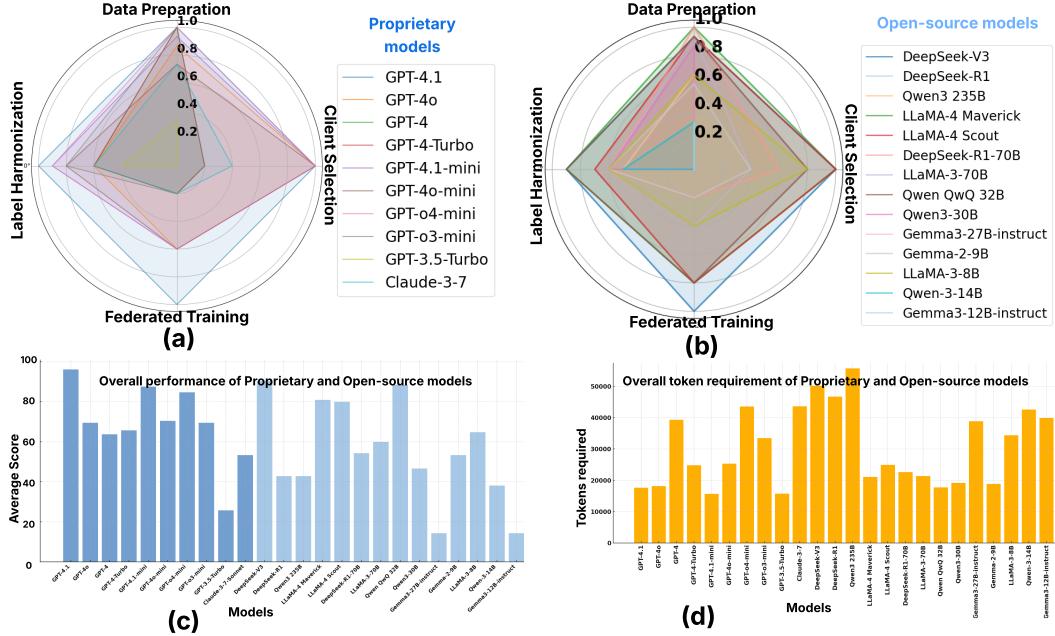


Figure 1: Performance of 24 LLM Agents on 4 FL sub-tasks over 6 healthcare environments. (a) and (b) show the performance of proprietary and open-source models respectively on four subtasks each, *viz.*, Client Selection, Data preprocessing, Label Harmonization, and Federated Training. (c) and (d) show the average score and mean overall token requirement of all models across all tasks.

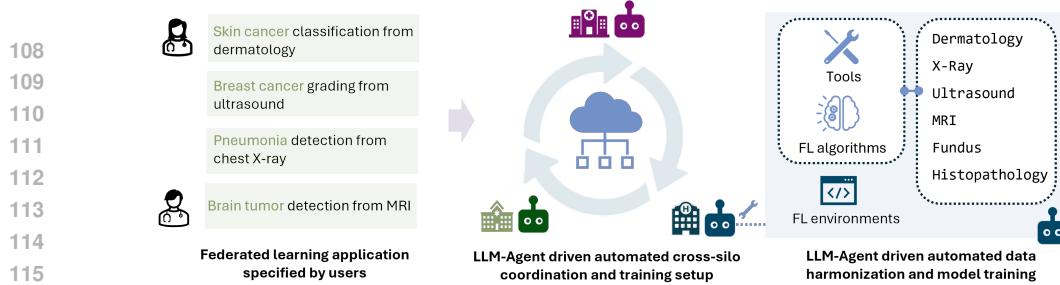
in FL initiatives despite having valuable local data. To this end, in this paper, we investigate the capabilities of LLM Agents in tackling these issues with minimal human intervention.

The rapid advancement of LLMs has led to the emergence of autonomous AI agents capable of executing complex, multi-step tasks across various domains (Gur et al.; Gou et al.; Cai et al.; Li et al., 2023a; Wang et al., 2023; Wu et al.; Mei et al., 2024; Chu et al., 2025; Qiu et al., 2024; Luo et al., 2025). This capability can be particularly transformative for real-world healthcare FL, where agent-based automation can reduce the operational burden on healthcare sites and enable broader participation in collaborative AI development. There are no existing works on agent-driven FL workflow; for general-purpose agents or agentic FL works, refer to **Related Works in Appendix A**.

To this end, we introduce an agentic FL framework (see Figs. 2 & 3) along with a benchmark **FedAgentBench** (see Fig. 1), designed to systematically evaluate the performance of LLM-driven agents in orchestrating FL workflows. To ensure comprehensive coverage, we incorporate 201 datasets, 6 major medical imaging modalities, and 40 representative FL algorithms designed for diverse real-world healthcare objectives and cross-client data compositions. To the best of our knowledge, this is the first work addressing FL problem-solving capabilities of LLM Agents directly dealing with server and client interactions. Our benchmark makes the following key contributions:

(1) Technical contribution: We first present a **plug-and-play modular agentic FL framework** supporting 40 FL algorithms and 24 LLM agents. It also allows for easy integration of new FL algorithms, agents and tasks with minimal adaptation. It is a unified FL framework with multi-faceted scenarios, multiple imaging modalities, and complex FL workflow structures. It encompasses four realistic and interlinked agent-driven FL phases: (i) **Client Selection**, where server and client agents communicate dataset suitability, (ii) **Data Preprocessing**, involving data restructuring, cleaning, and standardization using learned tools, (iii) **Label Harmonization**, where agents align inconsistent label taxonomies across clients, and (iv) **Federated Model Training**, where selected algorithms are deployed in a decentralized setup. It is worth noting that while we simulate healthcare environments in this work, the framework can be readily extended to other FL settings such as finance, IoT, etc.

(2) Dataset and Task contribution: To evaluate the effectiveness of LLM agents in real-world healthcare tasks, we construct a realistic simulation of inter-hospital collaboration within a FL framework in representative clinical scenarios. Specifically, we curate and publicly release **six medical imaging FL agentic environments** comprising a total of **201 datasets** and a diverse collection of tasks spanning a range of difficulties. To introduce greater variability across clients, we systematically modify the original image resolutions, file format extensions, and intensity distributions



108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161

Figure 2: Overview of our agent-driven FL setup. First, user defines task specification. Accordingly, LLM agents perform server-client coordination and complete required tasks using available tools and FL algorithms in any of the 6 modality-specific healthcare environments.

of the client datasets. Additionally, we carefully inject noisy and irrelevant samples spanning images from other modalities, text files, and other extraneous formats into client data directories to simulate realistic uncurated data environments and reflect the challenges of real-world clinical settings.

(3) Empirical contribution: As a part of FedAgentBench, we evaluate the performance of 24 LLM agents across diverse FL tasks based on task completion rate (*i.e.*, success rate), token efficiency, and overall time required. We investigate how varying levels of prompt granularity affect task execution and systematically compare agent performance across different autonomy tiers: guided tool invocation, autonomous planning, and fully independent script generation. Our analysis provides a comprehensive assessment of agentic capabilities and limitations in supporting real-world collaborative healthcare workflows. We will open-source and continuously update the benchmark on Github repository to support FL research and help healthcare data holders fully realize the value of cross-silo data.

Research Questions. FedAgentBench is designed around 5 central research questions that capture the core operational challenges faced by LLM agents in FL workflows (Detailed in §3.2 and 3.3):

RQ1: Are there particular phases of the FL workflow that are especially challenging for LLM agents? How does LLM agent performance vary across different phases of the workflow?

A1: Across 24 models, we observe a consistent difficulty hierarchy: *Label Harmonization > Data Preprocessing > Federated Training > Client Selection* with harmonization emerging as the dominant bottleneck due to its need for multi-hop semantic alignment across heterogeneous client taxonomies.

RQ2: What role does the granularity of prompts or instructions play in how reliably agents complete different steps of the workflow? A2: Fine-grained, structured prompts substantially increase success rates, especially for the complex semantic phases. By contrast, goal-oriented prompts often lead to reasoning drift, skipped steps, and hallucinated structures.

RQ3: To what extent can we rely on scale alone to predict how well an agent will perform? Does choosing a larger LLM translate into more dependable agent behaviour? A3: Empirically, model scale is not a reliable predictor of performance. Several mid-sized models (*e.g.*, Qwen QwQ-32B, LLaMA-4 Scout) outperform much larger models, indicating that instruction-following ability and architectural grounding outweigh parameter count.

RQ4: Do challenging real-world Federated Learning subtasks such as label harmonization and data preprocessing expose systematic weaknesses in current LLM agents? A4: Yes, these tasks consistently surface systematic failure modes including misaligned label mappings, multi-step workflow collapse, speculative reasoning, and poor grounding in tool outputs and workspace structure.

RQ5: How pronounced is the difference in performance between proprietary and open-source agents across the FL workflow? A5: The performance gap exists but is phase-dependent: proprietary models excel in the hardest stages (preprocessing, harmonization), while strong open-source models often match or exceed them in simpler stages (client selection, training initiation).

2 FEDAGENTBENCH FRAMEWORK

2.1 PROBLEM FORMULATION AND OVERVIEW

Given a user-defined task specification for federated medical image analysis, denoted as \mathcal{T} , our objective is to construct and execute a complete FL pipeline through collaborative decision-making by a set of autonomous agents. As outlined in Fig. 3, FedAgentBench consists of two main components:

162 **(i) Federated medical imaging workspace** \mathcal{W} which can be sub-categorized to server workspace
 163 \mathcal{W}_s and client workspace \mathcal{W}_c as well as **(ii) Multi-agent coordination system** \mathcal{A} . The workspace
 164 \mathcal{W} encapsulates the critical resources required for FL pipeline construction and includes: (1) client
 165 metadata files (data cards) containing natural language descriptions of local datasets (in \mathcal{W}_c), (2) FL
 166 algorithm specifications (in \mathcal{W}_s) and tool usage descriptions (in \mathcal{W}_c and \mathcal{W}_s) and (3) structured code
 167 templates for each phase of the FL workflow (in \mathcal{W}_c and \mathcal{W}_s).

168 Built on top of this workspace, the agents operate under a divide-and-conquer strategy to address
 169 the complexity and modularity of the entire FL process. The server-client agent system $\mathcal{A} =$
 170 $\{S_1, S_2, S_3, S_4, C_1, C_2, C_3\}$ comprises 7 role-specialized LLM agents (see Fig. 3) responsible
 171 for: (1) client selection and server-client communication or orchestration (S_1, S_2, C_1), (2) data
 172 preprocessing and cleaning (C_2), (3) label harmonization (C_3), and (4) federated model selection
 173 and training (S_3, S_4). The collaborative pipeline proceeds iteratively as agents can invoke tools,
 174 write scripts, or reason over workspace content to solve subtasks, with execution feedback enabling
 175 adaptation. This process can be formally represented as: $\{D_i, R_i\} = \mathcal{A}(D_{i-1}, R_{i-1}, \mathcal{T} \mid \mathcal{W})$
 176 where D_i denotes the code, decisions, or configurations generated or modified in the i -th iteration,
 177 and R_i represents execution results or tool feedback (e.g., logs, errors, evaluation metrics), with
 178 $D_0 = R_0 = \emptyset$. The goal is to produce a complete, executable FL pipeline satisfying task specification
 179 \mathcal{T} , measured in terms of success and efficiency under real-world constraints simulated by \mathcal{W} .

180 2.2 CLIENT DATASET CURATION AND FL ALGORITHM INTEGRATION

181 **Broad coverage of real-world medical specialties and data sets:** We construct FedAgentBench
 182 clients by adapting **201 publicly available datasets** with 2D and 3D dimensionality across 6 different
 183 medical imaging modalities *viz.* **25 Dermatology, 33 Ultrasound, 63 Fundus, 32 X-Ray, 28 MRI,**
 184 **and 20 Histopathology datasets.** It spans a broad range of tasks, including disease classification
 185 (e.g., tumor detection, cancer subtype identification), disease staging or grading (e.g., cancer and
 186 diabetic retinopathy severity levels), anatomical or pathological region segmentation (e.g., tumor
 187 or stroke localization), object detection, regression, reconstruction, *etc.* Each client is simulated to
 188 comprise one or more of these datasets, reflecting the diversity and heterogeneity typical of real-world
 189 healthcare institutions. We construct a datacard accompanying each client based on the metadata
 190 sourced from its original publication, repository or website. **See Appendix C.1 & Listings 6-8.**

191 **Cross-client data heterogeneity beyond distribution shifts:** In order to introduce greater variability
 192 across clients and better emulate the heterogeneity found in real-world clinical data silos, we
 193 systematically modify several aspects of the original datasets:

194 **(i) Structured Dataset Perturbations:** We introduce systematic modifications to dataset characteristics,
 195 such as varying image resolutions (e.g., downsampling images), altering file format extensions
 196 (e.g., converting `.png` files to `.jpeg`, `.bmp`, or `.tiff`), and modifying intensity distributions to
 197 reflect differences in scanner settings or preprocessing pipelines.

198 **(ii) Inclusion of Uncurated and Irrelevant Files:** To reflect the messiness of real-world clinical
 199 storage, we inject non-image and unrelated files into client directories. These include textual notes
 200 (`.txt`, `.doc`, `.pdf`), spreadsheets (`.csv`, `.xls`), and miscellaneous files (e.g., `.log`, `.xml`,
 201 `.ini`). For example, our dermatoscopy dataset contains lesion images mixed with dermatologist
 202 notes in `.pdf` format and other unrelated documents.

203 **(iii) Simulation of Label and Modality Noise:** We simulate common data quality issues by introducing
 204 random duplication of 2-5 samples, injecting 2-5 anatomically or modality-inconsistent images,
 205 and deliberately corrupting labels of 2-5 samples to model annotation noise in each dataset.

206 These artifacts challenge the robustness of agent-based preprocessing and reflect the complexities
 207 encountered in real hospital PACS or data repositories. See Appendix C for more details.

208 **Algorithm suite for a wide spectrum of FL settings:** As a part of the benchmark design, we also
 209 curate a comprehensive suite of **40 FL algorithms** by integrating and adapting existing implementations.
 210 This algorithm collection spans a broad spectrum of FL paradigms enabling standardized and
 211 reproducible evaluation across diverse medical imaging tasks (**See Appendix §C.4**). This includes:

212 **(i) Classical FL algorithms** such as FedAvg, FedProx, and Scaffold, which address baseline
 213 aggregation and client drift; **(ii) Personalized FL algorithms** like Per-FedAvg, pFedMe, and
 214 FedRep, which tailor models to heterogeneous client data distributions; **(iii) Regularization-based**
 215 **approaches** like Ditto which impose constraints to preserve global knowledge during local updates;
 216 **(iv) Knowledge Distillation-based methods** such as FedDF, enabling model-agnostic communica-

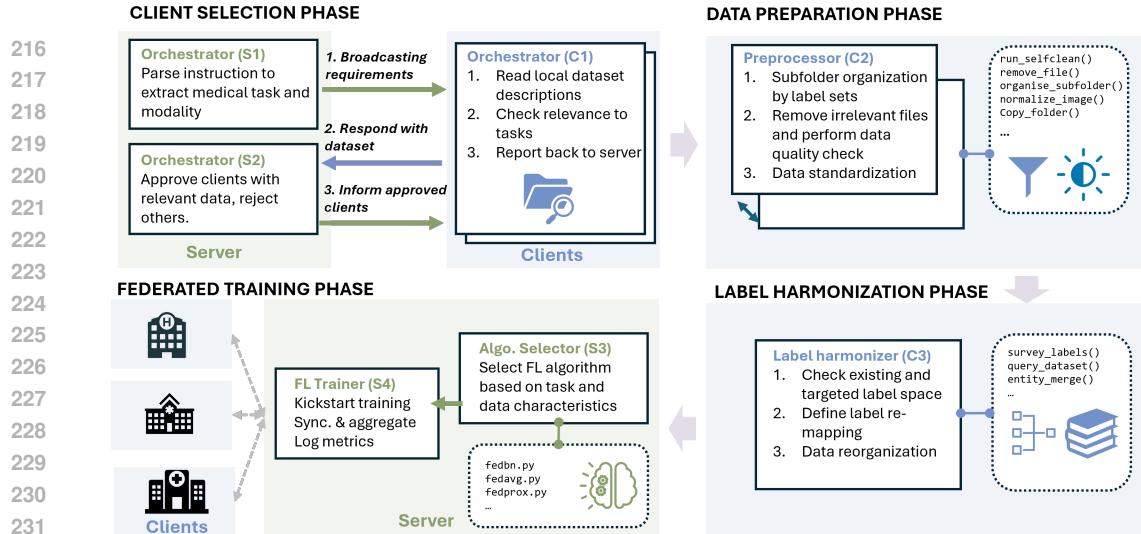


Figure 3: An overview of the FedAgentBench Framework. It comprises 7 role-specialized LLM agents ($S_1, S_2, S_3, S_4, C_1, C_2, C_3$) for completing 4 distinct phases of the FL workflow (see §2.3)

tion via logits; (v) **Domain generalization techniques** like FedSR, FedDG, and FedIRM, which aim to learn invariant representations across non-IID clients; and (vi) **Optimization and scheduling variants**, such as FedNova which address stability, and convergence rate.

2.3 FEDERATED AGENTIC FRAMEWORK CONSTRUCTION

FL workflows typically follow a common set of phases, from which we abstract the key human roles and tasks fundamental to their execution as discussed below (See Appendix B.2 for more details):

1. Client orchestrator agents: These agents act as the coordinators of the framework by communicating between the server and clients as well as by selecting the most suitable clients for the task based on the user requirements and individual client responses (see Fig. 4).

Server agent S_1 interprets the user-defined task \mathcal{T} and communicates imaging modality/task requirements to initiate client selection. For this, it first parses \mathcal{T} and broadcasts a query to all Client Agents (*i.e.*, healthcare sites). Each Client Agent C_1 reads local dataset description file, which contains metadata about available datasets, including label sets/imaging types. Based on semantic and modality matching, C_1 evaluates relevance of its datasets to \mathcal{T} , returning only matching datasets (if any). Server Agent S_2 collects these responses and selects a subset of relevant clients C_{sel} , which are then approved for further processing (see Figs. 9-14 in Appendix D).

2. Data pre-processor agent: It is responsible for preparing selected client datasets for effective participation in the FL pipeline. Given the diversity of data storage formats and quality issues across real-world sites, Data pre-processor agent C_2 at each client ensures that the dataset adheres to a standardized structure and meets minimum quality criteria. Concretely, it is responsible for standardizing and cleaning datasets at each selected client (see Fig. 5). This includes:

- (i) **Subfolder Organization:** Verifies whether datasets are organized into class-specific subfolders. If disorganized, C_2 restructures the folder hierarchy.
- (ii) **File Cleaning:** Removes irrelevant files (non-image formats `.txt`, `.csv` etc.) to ensure format consistency.
- (iii) **Data Cleaning:** Detects and flags duplicates, off-topic samples, and noisy labels, which are

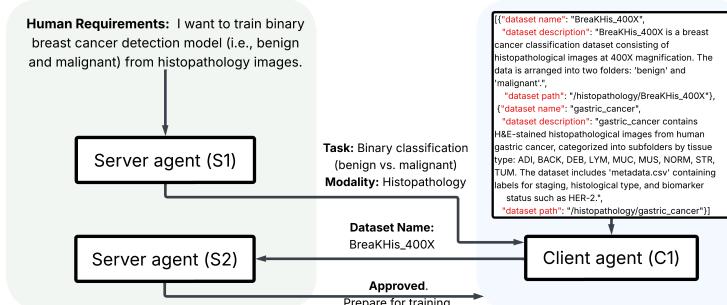


Figure 4: Client orchestrator agents S_1, C_1 , and S_2 in a histopathology-based breast cancer classification task

270 then removed. This ensures all selected clients have curated structurally consistent data, enabling
 271 downstream harmonization and consequent training (see **Figs. 35-36 in Appendix D**).
 272

273 **(iv) Data Normalization/Standardization:** Standardizes images across clients
 274 based on resolution, intensity, and file extension. This
 275 agent thus plays an essential
 276 role in bridging the gap be-
 277 tween raw, heterogeneous
 278 clinical data and the clean,
 279 harmonized inputs required
 280 for FL. Its operations ensure
 281 that all participating clients
 282 contribute structurally con-
 283 sistent, high-quality data
 284 harmonized across clients,
 285 which is crucial for the suc-
 286 cess of the overall FL system.
 287

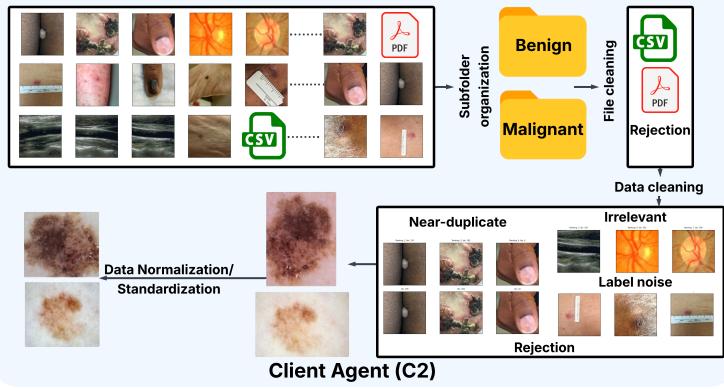


Figure 5: Data pre-processor agent C_2 in skin cancer detection task

289 **3. Task-conditioned label harmonizer agent:** This agent (C_3) addresses one of the most critical
 290 challenges in multi-institutional FL, *i.e.*, the inconsistency in label nomenclature and granularity
 291 across client datasets (see Fig. 6). Due to variations in annotation protocols, terminologies, and
 292 domain-specific taxonomies, class labels across clients may not align semantically or structurally.
 293 C_3 plays a pivotal role in reconciling these differences based on the user requirements: **(i) Class**
 294 **Inspection:** Enumerates all class labels present in client datasets.

295 **(ii) Label Mapping:** Converts fine-grained labels (e.g., "melanoma", "nevus") to harmonized
 296 classes (e.g., "malignant", "benign") according to a self-developed mapping schema.

297 **(iii) Data Reorganization:**
 298 Reorganizes the dataset
 299 structure to reflect har-
 300 monized labels, aligning
 301 image samples with their
 302 mapped class definitions.
 303 This standardization en-
 304 ables cross-client training
 305 without semantic conflicts
 306 in label interpretation.

307 Through these actions, the
 308 agent guarantees that all
 309 clients adhere to a shared la-
 310 bel vocabulary.

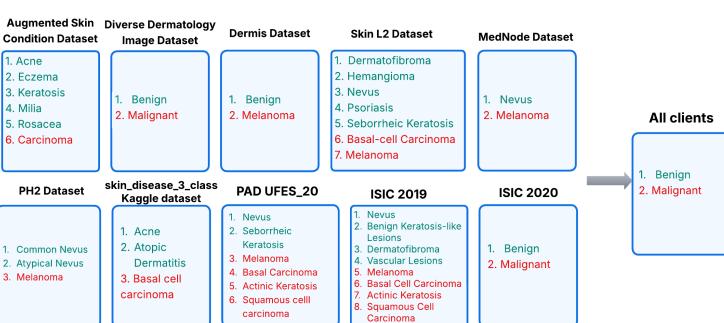


Figure 6: Label harmonization by agent C_3 in dermatology-based skin cancer detection (benign/malignant classes color-coded in green/red)

311 **4. Federated trainer agents:** These agents are responsible for initiating the actual federated training
 312 process across the selected set of clients and play a central role in converting the prepared environment
 313 into a functioning FL system. They initiate and coordinate federated training in 2 steps:

314 (i) Based on \mathcal{T} , **FL Algorithm Selector Agent** (S_3) queries a registry of 40 FL algorithms containing
 315 the algorithmic descriptions and then selects a suitable method (e.g., FedAvg, pFedSim, FedSR)
 316 based on user requirements.

317 (ii) **Trainer Agent** (S_4) then distributes training details to approved clients and executes Federated
 318 Training. During training, S_4 logs per-client and global metrics (e.g., accuracy) and performs
 319 model aggregation. Its modular structure supports plug-and-play experimentation with different FL
 320 algorithms and training configurations.

321 2.4 PRIVACY PRESERVING AND MODULAR DESIGN

322 A key advantage of our framework is its modular design across phases and agent specializations:
 323 Each agent component and phase can be independently evaluated, replaced, or extended. More

Table 1: Comparison of LLM agents in **Dermatology** environment based on skin cancer detection task. Here **P**, **R**, **F1** indicate Precision, Recall, and F1 score of **selected clients vs. the canonical eligible client set**. **S**, **D**, **F** indicate Schema Compliance Rate, Duplicate Removal Rate, and Format Normalization Rate. **E**, **C**, **Co** indicate Exact-match Accuracy, Coverage Rate, and Conflict Rate. **T** indicates Training-start verification score.

Model	Fine-grained guidance								Goal-oriented guidance									
	Client-Sel			Data-Pre		Label-Harm		Fed-Train		Client-Sel			Data-Pre		Label-Harm		Fed-Train	
	P, R, F1		S, D, F	E, C, Co		T		P, R, F1		S, D, F		E, C, Co		T		E, C, Co		T
Proprietary Models																		
GPT-4.1	0.96, 1.00, 0.98	1.00, 0.97, 1.00	0.61, 0.65, 0.35	0.99	0.88, 0.86, 0.87	1.00, 0.96, 0.98	0.61, 0.61, 0.39	0.85	0.79, 0.76, 0.77	0.96, 0.91, 0.92	0.16, 0.24, 0.76	0.18	0.70, 0.68, 0.69	0.05, 0.00, 0.00	0.00, 0.01, 0.96	0.43		
GPT-4o	0.88, 0.89, 0.88	1.00, 0.94, 0.95	0.18, 0.27, 0.73	0.21	0.91, 0.89, 0.90	0.22, 0.29, 0.71	0.61	0.88, 0.79, 0.83	1.00, 0.98, 0.97	0.25, 0.29, 0.71	0.45	0.50, 0.56, 0.53	1.00, 0.96, 0.98	0.23, 0.26, 0.74	0.40	0.91, 0.93, 0.90	0.63	
GPT-4	1.00, 0.92, 0.96	0.02, 0.01, 0.00	0.22, 0.29, 0.71	0.61	1.00, 1.00, 1.00	0.00, 0.93, 0.98	0.59, 0.65, 0.35	0.61	1.00, 0.97, 0.98	0.57, 0.53, 0.57	0.59, 0.60, 0.40	0.58	0.94, 0.91, 0.92	0.98, 0.95, 0.96	0.74, 0.70, 0.73	0.60		
GPT-4-Turbo	0.91, 0.89, 0.90	0.41, 0.33, 0.39	0.19, 0.24, 0.76	0.64	0.64, 0.61, 0.62	1.00, 0.92, 1.00	0.60, 0.63, 0.37	0.61	0.90, 0.80, 0.85	0.74, 0.70, 0.73	0.45, 0.50, 0.50	0.60	0.86, 0.89, 0.87	0.00, 0.00, 0.00	0.44, 0.50, 0.50	0.63		
GPT-4.1-mini	1.00, 1.00, 1.00	0.00, 0.00, 0.00	0.59, 0.65, 0.35	0.61	GPT-4o-mini	0.64, 0.61, 0.62	1.00, 0.92, 1.00	0.60, 0.63, 0.37	0.61	0.71, 0.77, 0.74	0.05, 0.00, 0.00	0.44, 0.50, 0.50	0.21	0.32, 0.35, 0.33	0.04, 0.00, 0.00	0.43, 0.38, 0.38	0.26, 0.32, 0.68	
GPT-4o-mini	0.64, 0.61, 0.62	1.00, 0.92, 1.00	0.60, 0.63, 0.37	0.61	GPT-3.5-mini	0.86, 0.89, 0.87	0.00, 0.00, 0.00	0.45, 0.49, 0.51	0.58	0.41, 0.30, 0.35	0.43, 0.38, 0.38	0.00, 0.00, 1.00	0.21	0.67, 0.68, 0.67	0.44, 0.42, 0.42	0.21, 0.27, 0.73	0.42	
GPT-3.5-Turbo	0.32, 0.35, 0.33	0.04, 0.00, 0.00	0.00, 0.03, 0.97	0.18	Claude-3.7-Sonnet	0.67, 0.68, 0.67	0.44, 0.42, 0.42	0.21, 0.27, 0.73	0.42	0.69, 0.69, 0.69	0.40, 0.38, 0.39	0.26, 0.32, 0.68	0.44					
Open-source Models																		
Huge Models																		
DeepSeek-V3	0.79, 0.78, 0.78	0.97, 0.96, 0.94	1.00, 1.00, 0.00	0.78	DeepSeek-R1	0.70, 0.65, 0.67	0.00, 0.00, 0.00	0.02, 0.08, 0.92	0.03	0.76, 0.75, 0.75	0.77, 0.73, 0.75	0.81, 0.83, 0.17	0.82	0.62, 0.68, 0.65	0.01, 0.00, 0.00	0.01, 0.01, 0.97	0.00	
Qwen3 235B	0.62, 0.68, 0.65	0.01, 0.00, 0.00	0.02, 0.09, 0.91	0.00	LLaMA-4 Maverick	0.65, 0.69, 0.67	0.98, 0.90, 0.97	0.57, 0.66, 0.34	0.37	0.64, 0.69, 0.66	0.98, 0.90, 0.94	0.65, 0.68, 0.32	0.62	0.75, 0.77, 0.76	1.00, 0.93, 0.95	0.66, 0.73, 0.27	0.41	
LLaMA-4 Scout	0.75, 0.77, 0.76	1.00, 0.93, 0.95	0.66, 0.73, 0.27	0.41	0.79, 0.80, 0.79	1.00, 0.95, 0.97	0.56, 0.64, 0.36	0.44										
Large Models																		
DeepSeek-R1-70B	0.71, 0.71, 0.71	0.00, 0.00, 0.00	0.02, 0.03, 0.95	0.19	LLaMA-3-70B	0.72, 0.65, 0.68	0.17, 0.11, 0.12	0.17, 0.20, 0.80	0.43	0.64, 0.72, 0.68	0.00, 0.00, 0.00	0.03, 0.09, 0.91	0.00					
Medium Models																		
Qwen QwQ 32B	0.94, 0.92, 0.93	1.00, 0.96, 1.00	0.87, 0.89, 0.11	0.84	Qwen3-30B	0.74, 0.68, 0.71	0.04, 0.04, 0.03	0.05, 0.06, 0.94	0.19	0.86, 0.93, 0.89	1.00, 0.97, 1.00	0.57, 0.65, 0.35	0.64	0.30, 0.38, 0.34	0.00, 0.00, 0.00	0.26, 0.34, 0.29	0.00, 0.00, 0.00	
Gemma3-27B	0.30, 0.38, 0.34	0.00, 0.00, 0.00	0.00, 0.03, 0.97	0.01	Gemma-2-9B	0.69, 0.67, 0.68	0.24, 0.15, 0.19	0.19, 0.23, 0.77	0.24	0.60, 0.72, 0.65	0.24, 0.15, 0.17	0.17, 0.21, 0.79	0.19	0.72, 0.65, 0.68	1.00, 0.92, 0.98	0.38, 0.44, 0.56	0.20	
LLaMA-3-8B	0.72, 0.65, 0.68	1.00, 0.92, 0.98	0.38, 0.44, 0.56	0.20	Qwen3-14B	0.70, 0.69, 0.69	0.04, 0.00, 0.04	0.06, 0.11, 0.89	0.02	0.59, 0.65, 0.62	0.00, 0.00, 0.00	0.03, 0.07, 0.93	0.04	0.38, 0.36, 0.37	0.00, 0.00, 0.00	0.00, 0.00, 0.00	0.06, 0.08, 0.92	
Gemma3-12B-instruct	0.38, 0.36, 0.37	0.00, 0.00, 0.00	0.00, 0.05, 0.95	0.05	Gemma3-12B-instruct	0.34, 0.37, 0.35	0.00, 0.00, 0.00	0.00, 0.00, 0.00	0.04									

importantly, this modularity enables future expansion of the benchmark and adaptation to diverse real-world scenarios. For instance, additional components simulating privacy/safety audits conducted by humans or AI can be seamlessly inserted between server and client agents or workflow phases, without the need for altering the existing workflow.

It is to be noted that our framework enforces data privacy by design, aligning fully with FL principles. We explicitly prevent agents from ever accessing or transmitting raw data, model weights, or sensitive metadata. The server receives approvals/configuration signals only, not images, so the agent layer never handles patient data. Instead, agents operate at orchestration layer only and exchange only predefined information (JSON configs, file paths, status signals). They do not have direct access to raw client data (e.g., patient images) or sensitive metadata and never transmit patient data or intermediate outputs externally. Training is invoked via a tool wrapper that runs locally per client; no raw data leaves clients at the agent layer, *i.e.*, federated training is triggered by the agent, but executed on local clients via tools. All data preprocessing and label harmonization also happen locally at clients. Eg: In label harmonization, the agent creates mapping logic, but the mapping execution and label replacement are performed entirely on the local client side.

3 EXPERIMENTS AND RESULTS

3.1 IMPLEMENTATION AND EVALUATION DETAILS

We utilize the LangGraph architecture (Langgraph, 2025) for agent construction and workflow graph compilation. Each agent is assigned a tailored toolset, drawn from our proposed suite of 16 tools (see Appendix B.1), with the selection guided by the agent’s specific role and the need to omit redundant or irrelevant functionalities. In order to assess the capabilities of existing LLM agents, we validate a total number of 24 models on the FedAgentBench datasets, including: (1) 10 representative proprietary LLMs: GPT 4.1, GPT-4o, GPT-4, GPT-4-Turbo, GPT 4.1-mini, GPT-4o-mini, GPT o4-mini, GPT o3-mini, GPT-3.5 Turbo, and Claude-3.7 Sonnet. (2) 14 state-of-the-art open-sourced LLMs ranging from 9B to 685B: LLaMA series models (LLaMA-4 Maverick, LLaMA-4 Scout, LLaMA-3 70B, LLaMA-3 8B), DeepSeek series models (DeepSeek-V3, Deepseek-R1, DeepSeek-R1-Distill-Llama-70B), Qwen series models (Qwen 3 235B, Qwen QwQ 32B, Qwen 3 30B, Qwen 3 14B) and Gemma series models (Gemma 3 27B Instruct, Gemma 3 12B Instruct, Gemma 2 9B Instruct). We utilize APIs from (OpenAI, 2025), (Groq, 2025), (Deep Infra, 2025).

Evaluation metrics: We evaluate the agentic performance using a total of **13 key metrics** in different steps of the FL workflow: **(1) For each step**, we use **Success Rate** over **5 runs** which is a binary indicator of task success/completion. It evaluates the ability of the multi-agent framework to generate executable outputs that satisfy the task requirements. **(2) For client selection step**, we use **Precision, Recall, and F1 score** of selected clients vs. the canonical eligible client set **(and not of model performance)**. **(3) For data pre-processing step**, we use **(i) Schema Compliance Rate**, *i.e.*, proportion of correctly structured folders/files, **(ii) Duplicate Removal Rate**, *i.e.*, proportion of duplicates removed, and **(iii) Format Normalization Rate**, *i.e.*, proportion of files correctly normalized (e.g., format, resolution). **(4) For label harmonization step**, we use: **(i) Exact-match Accuracy** of label mappings vs. the canonical schema, **(ii) Coverage Rate**, *i.e.*, proportion of local classes successfully mapped, **(iii) Conflict Rate**, *i.e.*, proportion of classes with ambiguous mappings. **(5) For federated training step**, we use **Training Start Verification** as the metric to determine whether the agent produces valid configuration files, initializes the training process, and logs the start signal. Besides, **for each step**, we also compute **(6) Time Spent in seconds** which denotes the duration required to complete the task **(see Appendix D & Table 16 for comparison of average time)**; and **(7) Token Requirement** which indicates the number of tokens involved **(see Fig. 1 (d) for comparison of token requirement)**.

Tasks: The benchmark is tested on six representative real-world clinical tasks across six major medical imaging modalities: (i) Skin cancer detection from dermatology images (Tables 1 and 10), (ii) Breast cancer detection from ultrasound (Table 11), (iii) Glaucoma detection from fundus imaging (Table 14), (iv) Pneumonia detection from chest X-ray (Table 15), (v) Brain tumor detection from MRI (Table 13), and (vi) Lymph-node metastasis detection from histopathology (Table 2).

3.2 MAIN RESULTS AND KEY INSIGHTS

We summarize the overall success scores of all agent cores over 6 modality specific environments with two types of guidance styles for prompting LLMs *viz.*, fine-grained guidance (explicit step-by-step instructions) and goal-oriented guidance (high-level task description) in Fig. 7. We also show detailed performance breakdown of Dermatology environment in Table 1 and Histopathology in Table 2. **For detailed results in all other environments, please see Appendix D & Tables 10-15** Also, see Fig 1 (d) for overall token requirements of each model.

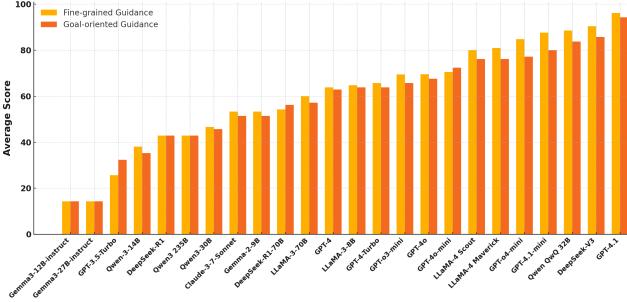


Figure 7: Overall performance of FedAgentBench

From the tables, we find proprietary models consistently outperform open-source ones across all FL stages. Besides, fine-grained guidance yields higher success rates than goal-oriented prompts for most models. Performance drops in more complex tasks like label harmonization compared to client selection. We also observe that model size alone does not guarantee performance (see Fig. 7). Instead, architectural design and instruction-following capability are more critical.

RQ1: Impact of Task Complexity: High success is observed in the initial and final steps of client orchestration and federated training across almost all agents, including weaker ones indicating that these tasks are relatively simpler. Data Pre-processing and Label Harmonization are seen to be major differentiators among agents. Weaker agents particularly fail to perform these tasks especially in goal-oriented scenarios, where planning and file structure comprehension are needed. Across almost all agents, label harmonization shows lowest success rates, regardless of guidance type. This suggests that aligning semantic labels across clients remains one of the hardest challenges. Among modalities, histopathology has the highest semantic complexity, potentially due to domain-specific terminology.

RQ2: Granularity of guidance: In fine-grained guidance, we provide explicit instruction to the models to follow a particular workflow whereas in goal-oriented guidance, we mention the overall objective of the agent without specifying the exact steps, thereby requiring autonomous planning or reasoning. Fine-grained guidance is seen to outperform goal-oriented guidance across almost every model, especially for weaker agents. More capable models like GPT-4.1 and DeepSeek-V3 close this gap, showing their capability to plan even based on implicit prompts.

Table 2: Comparison in terms of success rate over 5 runs for **Lymph-node metastasis detection** task in **Histopathology** environment

Model	Fine-grained guidance					Goal-oriented guidance				
	Client-Sel		Data-PreLabel-HarmFed-Train		Overall	Client-Sel		Data-PreLabel-HarmFed-Train		Overall
	S_1, C_1, S_2	C_2	C_3	S_3, S_4		S_1, C_1, S_2	C_2	C_3	S_3, S_4	
Proprietary Models										
GPT-4.1	5/5, 4/5, 5/5	5/5	5/5	4/5, 5/5	94.29	5/5, 4/5, 5/5	5/5	5/5	4/5, 5/5	94.29
GPT-4o	5/5, 0/5, 5/5	5/5	2/5	1/5, 5/5	65.71	5/5, 0/5, 5/5	5/5	1/5	1/5, 5/5	62.86
GPT-4	5/5, 1/5, 5/5	0/5	1/5	2/5, 5/5	54.29	5/5, 1/5, 5/5	0/5	0/5	2/5, 5/5	51.43
GPT-4-Turbo	5/5, 1/5, 5/5	1/5	1/5	2/5, 5/5	57.14	5/5, 1/5, 5/5	4/5	1/5	2/5, 5/5	65.71
GPT-4.1-mini	5/5, 3/5, 5/5	5/5	4/5	3/5, 5/5	85.71	5/5, 3/5, 5/5	3/5	4/5	3/5, 5/5	80.00
GPT-4o-mini	5/5, 1/5, 3/5	5/5	3/5	2/5, 4/5	65.71	5/5, 1/5, 3/5	5/5	1/5	2/5, 4/5	60.00
GPT-04-mini	5/5, 2/5, 5/5	5/5	3/5	2/5, 5/5	77.14	5/5, 2/5, 5/5	4/5	2/5	2/5, 4/5	68.57
GPT-03-mini	5/5, 5/5, 5/5	0/5	2/5	3/5, 5/5	71.43	5/5, 4/5, 5/5	0/5	2/5	3/5, 5/5	68.57
GPT-3.5-Turbo	5/5, 0/5, 0/5	0/5	0/5	1/5, 3/5	25.71	5/5, 0/5, 0/5	2/5	0/5	1/5, 3/5	31.43
Claude-3-7-Sonnet	5/5, 2/5, 3/5	2/5	1/5	2/5, 3/5	51.43	5/5, 2/5, 3/5	2/5	1/5	2/5, 5/5	57.14
Open-source Models										
Huge Models										
DeepSeek-V3	5/5, 3/5, 5/5	5/5	5/5	4/5, 5/5	91.43	5/5, 3/5, 5/5	4/5	5/5	4/5, 5/5	88.57
DeepSeek-R1	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86
Qwen3 235B	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86
LLaMA-4 Maverick	5/5, 2/5, 4/5	5/5	3/5	3/5, 5/5	77.14	5/5, 2/5, 4/5	5/5	3/5	3/5, 5/5	71.43
LLaMA-4 Scout	5/5, 2/5, 5/5	5/5	4/5	2/5, 5/5	80.00	5/5, 2/5, 5/5	5/5	3/5	2/5, 5/5	77.14
Large Models										
DeepSeek-R1-70B	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86
LLaMA-3-70B	5/5, 1/5, 5/5	1/5	1/5	1/5, 5/5	54.29	5/5, 1/5, 5/5	2/5	2/5	1/5, 5/5	60.00
Medium Models										
Qwen QwQ 32B	5/5, 4/5, 5/5	3/5	4/5	4/5, 5/5	85.71	5/5, 4/5, 5/5	2/5	4/5	4/5, 5/5	82.86
Qwen3-30B	5/5, 0/5, 5/5	0/5	0/5	1/5, 5/5	45.71	5/5, 0/5, 5/5	0/5	0/5	1/5, 5/5	45.71
Gemma3-27B-instruct	5/5, 0/5, 0/5	0/5	0/5	0/5, 0/5	14.29	5/5, 0/5, 0/5	0/5	0/5	0/5, 0/5	14.29
Small Models										
Gemma-2-9B	5/5, 1/5, 5/5	2/5	1/5	1/5, 5/5	57.14	5/5, 1/5, 5/5	1/5	1/5	1/5, 5/5	54.29
LLaMA-3-8B	5/5, 0/5, 5/5	5/5	2/5	1/5, 5/5	65.71	5/5, 0/5, 5/5	5/5	2/5	1/5, 5/5	65.71
Qwen-3-14B	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86	5/5, 0/5, 5/5	0/5	0/5	0/5, 4/5	40.00
Gemma3-12B-instruct	5/5, 0/5, 0/5	0/5	0/5	0/5, 0/5	14.29	5/5, 0/5, 0/5	0/5	0/5	0/5, 0/5	14.29

RQ3 & RQ5: Open Source Vs. Proprietary Models and Impact of Model Size:

Proprietary Model Performance: GPT-4.1 and GPT-4.1-mini show top-tier performance (85–100%), especially under fine-grained guidance. GPT-4o, although newer, struggles with label harmonization and federated training across all environments, leading to overall lower scores (62–71%). Claude-3.7-Sonnet achieves moderate performance (51–57%), inferior to GPT-4 variants. GPT-3.5-Turbo and older variants perform poorly, barely completing the complex stages.

Open-source Model Performance: We discuss agent performance based on model sizes below:

(i) Huge Models: DeepSeek-V3 is the strongest open-source model contender with 80–94% success rate comparable to the best proprietary models. Qwen3 and DeepSeek-R1 perform inconsistently, often failing in more structured stages like data pre-processing and label harmonization.

(ii) Medium and Large Models: Qwen QwQ 32B demonstrates strong performance (82–91%) and outperforms several proprietary models even under goal-oriented setups. LLaMA-4 Scout and Maverick also deliver competitive performance, especially in label harmonization and federated training, with scores in the 71–94% range. Other large models such as LLaMA-3-70B, and Qwen3-30B struggle with most tasks except initial client communication or final training step. Gemma3-27B-instruct is unusable under almost all these settings.

(iii) Small Models: Performance of 8-14B sized-models drops significantly. Most models (except LLaMA 3 8B) achieve less than or around 50% success. Particularly, Gemma 3-12B-instruct and Qwen 3 14B are observed to fail due to extreme hallucinations. These models are unable to perform any label-oriented reasoning and structured data operations, even under fine-grained instructions.

3.3 RQ4: AGENT FAILURE ANALYSIS:

We identify six key recurring failure modes of LLM agents across FL sub-tasks that highlight important limitations of current LLM capabilities in FL workflows (see Appendix D for more details):

(i) Lack of Domain-Specific Reasoning: The agents frequently fail to apply relevant medical domain knowledge. **Eg:** In label harmonization (Fig 6), the agents often miss subtle mismatches between dermatology folder names and coarse class labels possibly due to the lack of domain grounding and inability to handle naming conventions specific to medical datasets.

(ii) Failure in Multi-Step Planning: The agents are often unable to follow multi-step workflows, skipping essential operations where multiple sequential actions are required. **Eg:** Data pre-processor agents often overlook file/data cleaning steps of Fig. 5 due to multiple tasks in single execution cycle.

(iii) Overconfidence and Shortcutting: The agents recurrently provide wrong solutions, by defaulting to plausible but incorrect logic when unsure, instead of expressing uncertainty. **Eg:** Assigning both “nevus” and “melanoma metastasis” to the ‘benign’ class to simplify label mapping.

(iv) Hallucination in Structured Multi-Agent Tasks: The agents (particularly DeepSeek R1 and Gemma-based models) often generate irrelevant or unrelated outputs despite specific instructions

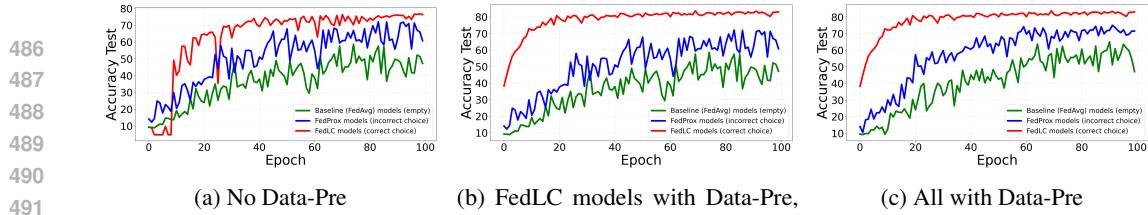


Figure 8: **Ablation analysis in Dermatology environment for instruction: Select FL algorithm that mitigates inter- and intra-client class imbalance.** Agents highlighted in red choose the correct method (FedLC), while those in blue select FedProx and others coded in green return no algorithm (defaulting to FedAvg). Subplots illustrate: (a) reduced performance when the data-preprocessing step fails, affecting all agentic systems; (b) improvement for agents in red that correctly preprocess; and (c) full performance gains when all agents successfully complete preprocessing.

due to misalignment with structured task formats and poor control over output scope (see Fig. 18-19 in Appendix D). **Eg:** When asked to select skin cancer dataset, Gemma-3 27B Instruct repeatedly returned philosophical or sarcastic monologues in foreign languages, tutorials on freelancing, etc.

(v) Task-Type and Modality Mismatch Due to Prior Assumptions: Agents can sometimes confuse tasks or ignore modality constraints due to frequency biases and shallow keyword matching instead of hierarchical task understanding. **Eg:** Recommending a malignant lesion segmentation dataset for a classification task or ultrasound datasets for histopathology-based breast cancer detection task.

(vi) Procedural Overthinking and Paralysis by Analysis: The reasoning/thinking agents often delay execution by speculating about dataset structure or missing dependencies without being asked, potentially due to excessive internal reasoning without grounding in file system or available information (see Fig. 16 in Appendix D). **Eg:** DeepSeek R1 repeatedly debates whether a client dataset should be selected without reading the dataset description file.

3.4 FINAL FEDERATED TRAINING PERFORMANCE:

To test whether agents truly select algorithms that improve overall FL performance, not just pass the “training-start” check, we run full end-to-end FL experiments. For the instruction: “*Train a federated learning model using an algorithm designed to mitigate both inter-client and intra-client class imbalance while still producing a strong global model*”, models like GPT-4.1, GPT-4o, GPT-4, Claude-3-Sonnet, DeepSeek-V3, Qwen QwQ 32B, Gemma-2-9B correctly select FedLC, while GPT-3.5-Turbo, Qwen3-235B, LLaMA-4 Maverick, LLaMA-4 Scout, and others wrongly choose FedProx. Some models viz., DeepSeek-R1, DeepSeek-R1-70B, Qwen3-30B, Qwen-3-14B return no algorithm and thus fall back to FedAvg. Across all runs, the performance ranks consistently as FedLC > FedProx > FedAvg, confirming that FedAgentBench captures real downstream impact rather than superficial setup success. See Fig. 8, Appendix Tables 18-20 for accuracy curves and detailed results.

4 CONCLUSION AND LIMITATION

In this paper, we introduced **the first agent-driven FL framework** and an associated benchmark, **FedAgentBench**, for evaluating LLM agents across diverse tasks constituting typical FL workflows. The evaluation covers 24 LLMs with varying sizes and a wide range of FL sub-tasks with varying difficulty levels in six modality-specific FL settings that closely simulate real-world clinical FL environments. Our framework is privacy preserving, comprehensive and modular. It includes 201 medical datasets and 40 FL algorithms and can be easily extended to incorporate more functionalities, settings, and algorithms specific to the user requirement. We investigated the impact of various factors like FL task complexity and granularity of guidance on the agent performance and analyzed the common failure modes of different agents. Our experiments reveal that across all environments, GPT-4.1 achieves almost perfect scores, under both fine-grained and goal-oriented prompting, whereas GPT-3.5-Turbo, Gemma3 series, and some Qwen variants consistently underperform across all stages and environments. DeepSeek-V3, Qwen QwQ 32B, and LLaMA-4 Maverick are the most reliable open-source agents across tasks. Unsurprisingly, fine-grained guidance consistently outperforms goal-oriented prompting, especially for less capable models. Our findings highlight that the order of complexity of the FL sub-tasks for most agents is: Label Harmonization > Data Pre-processing > Federated Training > Client Orchestration. Our experiments also show that larger model size does not necessarily correlate with better performance, i.e., some mid-sized models (30–40B) outperform larger ones (70B+). E.g., Qwen QwQ 32B consistently outperforms Qwen3-235B and DeepSeek-R1-70B.

540 REFERENCES
541542 3d-mri-ultrasound-brain dataset. <https://www.kaggle.com/datasets/shubhamcodez/3d-mri-ultrasound-brain-images>. Accessed: 2025-05-22.544 Buet dataset. <https://www.kaggle.com/datasets/jarintasnim090/buet-breast-ultrasound-data>. Accessed: 2025-05-22.545 546 Bus-uclm dataset. <https://www.kaggle.com/datasets/orvile/bus-uclm-breast-ultrasound-dataset,a>. Accessed: 2025-05-22.547 548 Busi dataset. <https://www.kaggle.com/datasets/sabahesaraki/breast-ultrasound-images-dataset,b>. Accessed: 2025-05-22.549 550 Breast-cancer dataset. <https://www.kaggle.com/datasets/aryashah2k/breast-ultrasound-images-dataset,a>. Accessed: 2025-05-22.551 552 Breast-ultrasound-images dataset. <https://www.kaggle.com/datasets/aryashah2k/breast-ultrasound-images-dataset,note = Accessed: 2025-05-22, b>.553 554 Camus-human-heart-data. <https://www.kaggle.com/datasets/shoybhasan/camus-human-heart-data>. Accessed: 2025-05-22.555 556 Ct2us dataset. <https://www.kaggle.com/datasets/siatsyx/ct2usforkidneyseg>. Accessed: 2025-05-22.557 558 Carotid-ultrasound-image dataset. <https://www.kaggle.com/datasets/orvile/carotid-ultrasound-images,a>. Accessed: 2025-05-22.559 560 Carotid-artery dataset. <https://www.kaggle.com/datasets/pahunichoudhary/carotid-artery-ultrasound-and-color-doppler,b>. Accessed: 2025-05-22.561 562 Ddti dataset. <https://www.kaggle.com/datasets/dasmehdixtr/ddti-thyroid-ultrasound-images>. Accessed: 2025-05-22.563 564 Dr dataset. <https://www.kaggle.com/c/diabetic-retinopathy-detection/data>. Accessed: 2025-05-22.565 566 Dermatology-tabular-dataset. <https://www.kaggle.com/datasets/olcaybolat1/dermatology-dataset-classification,a>. Accessed: 2025-05-22.567 568 Dermis dataset. <https://www.kaggle.com/datasets/farhatullah8398/skin-lesion-dermis-dataset,b>. Accessed: 2025-05-22.569 570 Dermnet dataset. <https://www.kaggle.com/datasets/shubhamgoel27/dermnet,c>. Accessed: 2025-05-22.571 572 Dermquest dataset. <http://www.dermquest.com>, d. Accessed: 2025-05-22.573 574 Fetal-head-us dataset. <https://www.kaggle.com/datasets/ankit8467/fetal-head-ultrasound-dataset-for-image-segment,a>. Accessed: 2025-05-22.575 576 Fetal-health-classification dataset. <https://www.kaggle.com/datasets/andrewmvd/fetal-health-classification,b>. Accessed: 2025-05-22.577 578 Isic2020 dataset. <https://kaggle.com/competitions/siim-isic-melanoma-classification,a>. Accessed: 2025-05-22.579 580 Isic2024 dataset. <https://kaggle.com/competitions/isic-2024-challenge,b>. Accessed: 2025-05-22.581 582 Liver-histopathology dataset. <http://kaggle.com/datasets/vibhingupta028/liver-histopathology-fibrosis-ultrasound-images>. Accessed: 2025-05-22.

594 Mednode dataset. <https://www.kaggle.com/datasets/prabhavsanga/med-node>.
 595 Accessed: 2025-05-22.
 596

597 Monkeypox-skin-image-dataset. <https://www.kaggle.com/datasets/nafin59/monkeypox-skin-lesion-dataset>. Accessed: 2025-05-22.
 598

599 Pcos dataset. <https://www.kaggle.com/datasets/anaghachoudhari/pcos-detection-using-ultrasound-images>. Accessed: 2025-05-22.
 600

602 Ph2dataset. <https://www.kaggle.com/datasets/spacesurfer/ph2-dataset>. Accessed: 2025-05-22.
 603

604 Rsna-asnr dataset. <https://www.kaggle.com/c/rsna-miccai-brain-tumor-radiogenomic-classification>. Accessed: 2025-05-22.
 605

607 Regensburg pediatric appendicitis dataset. <https://www.kaggle.com/datasets/joebeachcapital/regensburg-pediatric-appendicitis>. Accessed: 2025-05-22.
 608

610 Robotic_handheld_lumbar_spine_usdataset.. Accessed: 2025-05-22.
 611

612 Stare dataset. <https://www.kaggle.com/datasets/vidheeshnacode/stare-dataset>. Accessed: 2025-05-22.
 613

615 Thyroid_ultrasounddataset.. Accessed: 2025-05-22.
 616

617 Us3m dataset. <https://www.kaggle.com/datasets/timesxy/multimodal-breast-ultrasound-dataset-us3m>. Accessed: 2025-05-22.
 618

619 abdomen-mri dataset. <https://www.kaggle.com/datasets/imtkaggleteam/abdomen-mri-object-detection>. Accessed: 2025-05-22.
 620

621 alzheimer-mri dataset. <https://www.kaggle.com/datasets/borhanittrash/alzheimer-mri-disease-classification-dataset>. Accessed: 2025-05-22.
 622

624 Augmented skin conditions image dataset, howpublished = <https://www.kaggle.com/datasets/syedalinaqvi/augmented-skin-conditions-image-dataset>, note = Accessed: 2025-05-22.
 625

627 brain-cancer dataset. <https://www.kaggle.com/datasets/orvile/brain-cancer-mri-dataset>, a. Accessed: 2025-05-22.
 628

630 brain-mri-images dataset. <https://www.kaggle.com/datasets/ashfakyeafi/brain-mri-images>, b. Accessed: 2025-05-22.
 631

632 brain-tumor dataset. <https://www.kaggle.com/datasets/ultralytics/brain-tumor>, c. Accessed: 2025-05-22.
 633

635 brain-tumor-classification dataset. <https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri>, d. Accessed: 2025-05-22.
 636

637 brain-tumor-detection dataset. <https://www.kaggle.com/datasets/abhranta/brain-tumor-detection-mri>, e. Accessed: 2025-05-22.
 638

640 chasedb1 dataset. <https://blogs.kingston.ac.uk/retinal/chasedb1/>. Accessed: 2025-05-22.
 641

642 dermatologic-ultrasound dataset. <https://www.kaggle.com/datasets/alfageme/dermatologic-ultrasound-images>. Accessed: 2025-05-22.
 643

644 facial-mr dataset. <https://www.kaggle.com/datasets/humanaizedata/facial-mri-dataset-boost-your-ai-models>. Accessed: 2025-05-22.
 645

647 fallmud dataset. <https://www.kaggle.com/datasets/angeliqueloesch/fallmud>. Accessed: 2025-05-22.
 648

648 fetalultrasoundbrain dataset. <https://www.kaggle.com/datasets/rahimalargo/fetalultrasoundbrain>. Accessed: 2025-05-22.
 649
 650
 651 fhms-ultrasound dataset. <https://www.kaggle.com/datasets/jai132/fhms-ultrasound-dataset>. Accessed: 2025-05-22.
 652
 653 foraminal-stenosis dataset. <https://www.kaggle.com/datasets/axondata/foraminal-stenosis-mri-dataset>. Accessed: 2025-05-22.
 654
 655 glioma dataset. <https://www.kaggle.com/datasets/azharsaleem/mri-based-glioma-detection-dataset-with-masks>. Accessed: 2025-05-22.
 656
 657
 658 hbfmid dataset. <https://www.kaggle.com/datasets/orvile/human-bone-fractures-image-dataset-hbfmid>. Accessed: 2025-05-22.
 659
 660
 661 heart-mri dataset. <https://www.kaggle.com/datasets/adarshsng/heart-mri-image-dataset-left-atrial-segmentation>. Accessed: 2025-05-22.
 662
 663 hippocampal dataset. <https://www.kaggle.com/datasets/aryashah2k/hippocampal-sparring-dataset>. Accessed: 2025-05-22.
 664
 665
 666 indian-diabetic-retinopathy dataset. <https://www.kaggle.com/datasets/aaryapatel98/indian-diabetic-retinopathy-image-dataset>. Accessed: 2025-05-22.
 667
 668
 669 isles dataset. <https://www.kaggle.com/datasets/orvile/isles-2022-brain-stroke-dataset>. Accessed: 2025-05-22.
 670
 671
 672 Fetal_{health}classificationdataset.. Accessed: 2025-05-22.
 673
 674 mendeley_{breast}dataset.. Accessed: 2025-05-22.
 675
 676 mri-and-pet dataset. <https://www.kaggle.com/datasets/grantmcnatt/mri-and-pet-dice-similarity-dataset>. Accessed: 2025-05-22.
 677
 678 multimodal-ultrasound-vascular-segmentation dataset. <https://www.kaggle.com/datasets/among22/multimodal-ultrasound-vascular-segmentation>. Accessed: 2025-05-22.
 679
 680
 681 oasis-1 dataset. <https://www.kaggle.com/datasets/mdfahimbinamin/oasis-1-fastsurfer-quickseg-segmentation-dataset>. Accessed: 2025-05-22.
 682
 683 phantom dataset. <https://www.kaggle.com/datasets/ukeppendorf/frequently-traveling-human-phantom-fthp-dataset>. Accessed: 2025-05-22.
 684
 685
 686 pmram dataset. <https://www.kaggle.com/datasets/orvile/pmram-bangladeshi-brain-cancer-mri-dataset>. Accessed: 2025-05-22.
 687
 688
 689 prostate-annotated dataset. <https://www.kaggle.com/datasets/haithem1999/prostate-annotated-dataset-for-image-segmentation,a>. Accessed: 2025-05-22.
 690
 691
 692 prostate-mri-us dataset. <https://www.kaggle.com/datasets/dsptlp/prostate-mri-us-biopsy,b>. Accessed: 2025-05-22.
 693
 694
 695 refuge2 dataset. <https://www.kaggle.com/datasets/victorlemosml/refuge2>. Accessed: 2025-05-22.
 696
 697
 698 skin_{disease}clsaggledataset., a. Accessed: 2025-05-22.
 699
 700 skin_{infection}dataset., b. Accessed: 2025-05-22.
 701
 702 spinal-cord-dataset. <https://www.kaggle.com/datasets/trainingdatapro/spinal-cord-dataset>. Accessed: 2025-05-22.

702 strokehead_{mri}dataset.. Accessed: 2025-05-22.
 703

704 ultra-lr-hr-ultrasound dataset. <https://www.kaggle.com/datasets/chirag2466/ultra-lr-hr-ultrasound-image-dataset-for-research>, a. Accessed: 2025-05-22.
 705

706

707 ultrasound-nerve-segmentation dataset. <https://www.kaggle.com/competitions/ultrasound-nerve-segmentation>, b. Accessed: 2025-05-22.
 708

709

710 wmh-dataset. <https://www.kaggle.com/datasets/farahmo/wmh-dataset>. Accessed: 2025-05-22.
 711

712 Mohammad Amin Abbasi, Farnaz Sadat Mirnezami, and Hassan Naderi. Hamraz: A culture-based persian conversation dataset for person-centered therapy using llm agents. *arXiv preprint arXiv:2502.05982*, 2025.
 713

714

715 Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N. Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization, 2021. URL <https://arxiv.org/abs/2111.04263>.
 716

717

718 Vaibhav Aggarwal, Ojasv Kamal, Abhinav Japesh, Zhijing Jin, and Bernhard Schölkopf. Dars: Dynamic action re-sampling to enhance coding agent performance by adaptive tree traversals. *arXiv preprint arXiv:2503.14269*, 2025.
 719

720

721

722 Walid Al-Dhabyani, Mohammed Gomaa, Hussien Khaled, and Aly Fahmy. Dataset of breast ultrasound images. *Data in Brief*, 28:1–5, 2020.
 723

724

725 Alibabaei78. Ebhi segmentation dataset. <https://www.kaggle.com/datasets/alibabaei78/ebhi-seg>, 2024. Accessed: 2024-05-22.
 726

727

728 Rodolfo Stoffel Antunes, Cristiano André da Costa, Arne Küderle, Imrana Abdullahi Yari, and Björn Eskofier. Federated learning for healthcare: Systematic review and architecture proposal. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4):1–23, 2022.
 729

730

731 Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers, 2019. URL <https://arxiv.org/abs/1912.00818>.
 732

733

734 Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program Synthesis with Large Language Models, August 2021. URL <http://arxiv.org/abs/2108.07732>. arXiv:2108.07732 [cs].
 735

736

737

738 Reza Averly, Frazier N Baker, and Xia Ning. Liddia: Language-based intelligent drug discovery agent. *arXiv preprint arXiv:2502.13959*, 2025.
 739

740

741 Beosup. Lung segment dataset. <https://www.kaggle.com/datasets/beosup/lung-segment>, 2023. Accessed: 2025-05-22.
 742

743

744 Olivier Bernard 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.
 745

746

747 Matt Berseth. Isic 2017-skin lesion analysis towards melanoma detection. *arXiv preprint arXiv:1703.00523*, 2017.
 748

749 Michał Byra, Grzegorz Styczynski, Cezary Szmigelski, Piotr Kalinowski, Łukasz Michałowski, Rafał Paluszakiewicz, Bogna Ziarkiewicz-Wróblewska, Krzysztof Zieniewicz, Piotr Sobieraj, and Andrzej Nowicki. Transfer learning with deep convolutional neural network for liver steatosis assessment in ultrasound images. *International journal of computer assisted radiology and surgery*, 13(12):1895–1903, 2018.
 750

751

752

753

754

755 Tianle Cai, Xuezhi Wang, Tengyu Ma, Xinyun Chen, and Denny Zhou. Large language models as tool makers. In *The Twelfth International Conference on Learning Representations*.

756 John Capocyan. Cellnet beta version. <https://www.kaggle.com/datasets/johncapocyan/cellnet-beta-version>, 2024. Accessed: 2024-05-22.

757

758

759 Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. Unleashing the potential of
760 prompt engineering for large language models. *Patterns*, 2025a.

761

762 Dengsheng Chen, Jie Hu, Vince Junkai Tan, Xiaoming Wei, and Enhua Wu. Elastic aggregation
763 for federated optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and*
764 *Pattern Recognition (CVPR)*, pp. 12187–12197, June 2023a.

765

766 Hao-Yuan Chen, Cheng-Pong Huang, and Jui-Ming Yao. Verbal process supervision elicits better
767 coding agents. *arXiv preprint arXiv:2503.18494*, 2025b.

768

769 Hong-You Chen and Wei-Lun Chao. On bridging generic and personalized federated learning for
770 image classification, 2022. URL <https://arxiv.org/abs/2107.00778>.

771

772 Junying Chen, Chi Gui, Anningzhe Gao, Ke Ji, Xidong Wang, Xiang Wan, and Benyou Wang. Cod,
773 towards an interpretable medical agent using chain of diagnosis. *arXiv preprint arXiv:2407.13301*,
2024.

774

775 Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared
776 Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen
777 Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray,
778 Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter,
779 Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth
780 Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang,
781 Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N.
782 Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles
783 Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish,
784 Ilya Sutskever, and Wojciech Zaremba. Evaluating Large Language Models Trained on Code, July
2021. URL <http://arxiv.org/abs/2107.03374>. arXiv:2107.03374 [cs].

785

786 Yiqiang Chen, Wang Lu, Xin Qin, Jindong Wang, and Xing Xie. Metafed: Federated learning
787 among federations with cyclic knowledge distillation for personalized healthcare, 2023b. URL
788 <https://arxiv.org/abs/2206.08516>.

789

790 Zhaoling Chen, Xiangru Tang, Gangda Deng, Fang Wu, Jialong Wu, Zhiwei Jiang, Viktor Prasanna,
791 Arman Cohan, and Xingyao Wang. Locagent: Graph-guided llm agents for code localization. *arXiv*
792 *preprint arXiv:2503.09089*, 2025c.

793

794 Zhen Chen, Zhihao Peng, Xusheng Liang, Cheng Wang, Peigan Liang, Linsheng Zeng, Minjie Ju,
795 and Yixuan Yuan. Map: Evaluation and multi-agent enhancement of large language models for
inpatient pathways. *arXiv preprint arXiv:2503.13205*, 2025d.

796

797 Hojun Cho, Donghu Kim, Soyoung Yang, Chan Lee, Hunjoo Lee, and Jaegul Choo. Building
798 resource-constrained language agents: A korean case study on chemical toxicity information. *arXiv*
799 *preprint arXiv:2503.17753*, 2025.

800

801 Zhendong Chu, Shen Wang, Jian Xie, Tinghui Zhu, Yibo Yan, Jinheng Ye, Aoxiao Zhong, Xuming
802 Hu, Jing Liang, Philip S Yu, et al. Llm agents for education: Advances and applications. *arXiv*
803 *preprint arXiv:2503.11733*, 2025.

804

805 Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman,
806 Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. Skin lesion analysis
807 toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration
(isic). *arXiv preprint arXiv:1902.03368*, 2019.

808

809 Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared repre-
810 sentations for personalized federated learning, 2023. URL <https://arxiv.org/abs/2102.07078>.

810 Marc Combalia, Noel C. F. Codella, Veronica Rotemberg, Brian Helba, Veronica Vilaplana, Ofer Re-
 811 iter, Cristina Carrera, Alicia Barreiro, Allan C. Halpern, Susana Puig, and Josep Malvehy. Bcn20000:
 812 Dermoscopic lesions in the wild, 2019. URL <https://arxiv.org/abs/1908.02288>.
 813

814 Roxana Daneshjou, Kailas Vodrahalli, Roberto A Novoa, Melissa Jenkins, Weixin Liang, Veronica
 815 Rotemberg, Justin Ko, Susan M Swetter, Elizabeth E Bailey, Olivier Gevaert, et al. Disparities
 816 in dermatology ai performance on a diverse, curated clinical image set. *Science advances*, 8(31):
 817 eabq6147, 2022.

818 P. K. Darabi. Bone break classification image dataset. <https://www.kaggle.com/datasets/pkdarabi/bone-break-classification-image-dataset>, 2023. Accessed: 2025-05-22.
 819

820 Sergio MM de Faria, Jose N Filipe, Pedro MM Pereira, Luis MN Tavora, Pedro AA Assuncao,
 821 Miguel O Santos, Rui Fonseca-Pinto, Felicidade Santiago, Victoria Dominguez, and Martinha
 822 Henrique. Light field image dataset of skin lesions. In *2019 41st Annual International Conference of
 823 the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 3905–3908. IEEE, 2019.
 824

825 Coen De Vente, Koenraad A Vermeer, Nicolas Jaccard, He Wang, Hongyi Sun, Firas Khader, Daniel
 826 Truhn, Temirgali Aimyshev, Yerkebulan Zhanibekuly, Tien-Dung Le, et al. Airogs: Artificial
 827 intelligence for robust glaucoma screening challenge. *IEEE transactions on medical imaging*, 43(1):
 828 542–557, 2023.

829 Etienne Decenciere, Guy Cazuguel, Xiwei Zhang, Guillaume Thibault, J-C Klein, Fernand Meyer,
 830 Beatriz Marcotegui, Gwénolé Quellec, Mathieu Lamard, Ronan Danno, et al. Teleophta: Machine
 831 learning and image processing methods for teleophthalmology. *Irbm*, 34(2):196–203, 2013.
 832

833 Deep Infra. Deepinfra models documentation, 2025. URL <https://deepinfra.com/docs/models>. Accessed: 2025-05-16.

834 Qixin Deng, Qikai Yang, Ruibin Yuan, Yipeng Huang, Yi Wang, Xubo Liu, Zeyue Tian, Jiahao Pan,
 835 Ge Zhang, Hanfeng Lin, et al. Composerx: Multi-agent symbolic music composition with llms.
 836 *arXiv preprint arXiv:2404.18081*, 2024.

837 Yihe Deng and Paul Mineiro. Flow-dpo: Improving llm mathematical reasoning through online
 838 multi-agent learning. *arXiv preprint arXiv:2410.22304*, 2024.

839 Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated
 840 learning, 2020. URL <https://arxiv.org/abs/2003.13461>.
 841

842 Canh T. Dinh, Nguyen H. Tran, and Tuan Dung Nguyen. Personalized federated learning with
 843 moreau envelopes, 2022. URL <https://arxiv.org/abs/2006.08848>.
 844

845 Tuan Le Dinh. Monuseg 2018. <https://www.kaggle.com/datasets/tuanledinh/monuseg2018>, 2024. Accessed: 2024-05-22.

846 Wei Dong. The ann arbor architecture for agent-oriented programming. *arXiv preprint
 847 arXiv:2502.09903*, 2025.

848 Zhuoyun Du, Lujie Zheng, Renjun Hu, Yuyang Xu, Xiawei Li, Ying Sun, Wei Chen, Jian Wu, Haolei
 849 Cai, and Haohao Ying. Llms can simulate standardized patients via agent coevolution. *arXiv preprint
 850 arXiv:2412.11716*, 2024.

851 Gaurav Dutta. Fracturefusion: A symphony of bone breaks. <https://www.kaggle.com/datasets/gauravduttakait/fracturefusion-a-symphony-of-bone-breaks>,
 852 2023. Accessed: 2025-05-22.

853 Sachi Dwivedi. Kmc kidney histopathology dataset. <https://www.kaggle.com/datasets/sachidwivedi1234/kmc-kidney-histopathology-dataset2>, 2024. Accessed:
 854 2024-05-22.

855 factory.ai. Code Droid Technical Report, June 2024. URL <https://www.factory.ai/news/code-droid-technical-report>.

864 Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning
 865 with theoretical guarantees: A model-agnostic meta-learning approach. In H. Larochelle,
 866 M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural In-*
 867 *formation Processing Systems*, volume 33, pp. 3557–3568. Curran Associates, Inc.,
 868 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/24389bfe4fe2eba8bf9aa9203a44cdad-Paper.pdf.

870 Adibvafa Fallahpour, Jun Ma, Alif Munim, Hongwei Lyu, and BO WANG. Medrax: Medical
 871 reasoning agent for chest x-ray. In *Forty-second International Conference on Machine Learning*.

872

873 Sorouralsadat Fatemi and Yuheng Hu. Enhancing financial question answering with a multi-agent
 874 reflection framework. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pp.
 875 530–537, 2024.

876

877 George Fatouros, Kostas Metaxas, John Soldatos, and Manos Karathanassis. Marketsenseai 2.0:
 878 Enhancing stock analysis through llm agents. *arXiv preprint arXiv:2502.00415*, 2025.

879

880 Jared Feng. Histopath-sn. <https://www.kaggle.com/datasets/jaredfeng/histopathsn>, 2024. Accessed: 2024-05-22.

881

882 Jinghao Feng, Qiaoyu Zheng, Chaoyi Wu, Ziheng Zhao, Ya Zhang, Yanfeng Wang, and Weidi Xie.
 883 M^3builder: A multi-agent system for automated machine learning in medical imaging. *arXiv*
 884 *preprint arXiv:2502.20301*, 2025.

885

886 Forderation. Breakhis 400x. <https://www.kaggle.com/datasets/forderation/breakhis-400x>, 2024. Accessed: 2024-05-22.

887

888 Francisco Fumero, Silvia Alayón, José L Sanchez, Jose Sigut, and M Gonzalez-Hernandez. Rim-one:
 889 An open retinal image database for optic nerve evaluation. In *24th International Symposium on*
 890 *Computer-Based Medical Systems (CBMS)*, pp. 1–6. IEEE, 2011.

891

892 Fatemeh Ghezloo, Mehmet Saygin Seyfioglu, Rustin Soraki, Wisdom O Ikezogwo, Beibin Li, Tejoram
 893 Vivekanandan, Joann G Elmore, Ranjay Krishna, and Linda Shapiro. Pathfinder: A multi-modal
 894 multi-agent system for medical diagnostic decision-making applied to histopathology. *arXiv preprint*
 895 *arXiv:2502.08916*, 2025.

896

897 Arsham Gholamzadeh Khoei, Shuai Wang, Yinan Yu, Robert Feldt, and Dhasarathy Parthasarathy.
 898 GateLens: A reasoning-enhanced llm agent for automotive software release analytics. *arXiv e-prints*,
 899 pp. arXiv–2503, 2025.

900

901 Mohamed Gobara. Multi-class knee osteoporosis x-ray
 902 dataset. <https://www.kaggle.com/datasets/mohamedgobara/multi-class-knee-osteoporosis-x-ray-dataset>, 2023a. Accessed: 2025-05-22.

903

904 Mohamed Gobara. Osteoporosis database. <https://www.kaggle.com/datasets/mohamedgobara/osteoporosis-database>, 2023b. Accessed: 2025-05-22.

905

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

909

910 Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Minlie Huang, Nan Duan, Weizhu Chen,
 911 et al. Tora: A tool-integrated reasoning agent for mathematical problem solving. In *The Twelfth*
 912 *International Conference on Learning Representations*.

913

914 Dennis Grinwald, Philipp Wiesner, and Shinichi Nakajima. Federated learning over connected modes,
 915 2025. URL <https://arxiv.org/abs/2403.03333>.

916

917 Fabian Gröger, Simone Lionetti, Philippe Gottfrois, Alvaro Gonzalez-Jimenez, Matthew Groh,
 918 Roxana Daneshjou, Alexander A Navarini, Marc Pouly, Labelling Consortium, et al. Towards reliable
 919 dermatology evaluation benchmarks. In *Machine Learning for Health (ML4H)*, pp. 101–128. PMLR,
 920 2023.

918 Fabian Gröger, Simone Lionetti, Philippe Gottfrois, Alvaro Gonzalez-Jimenez, Ludovic Am-
 919 ruthalingam, Matthew Groh, Alexander Navarini, and Marc Pouly. Intrinsic self-supervision for data
 920 quality audits. *Advances in Neural Information Processing Systems*, 37:92273–92316, 2024.

921

922 Fabian Gröger, Simone Lionetti, Philippe Gottfrois, Alvaro Gonzalez-Jimenez, Ludovic Am-
 923 ruthalingam, Elisabeth Victoria Goessinger, Hanna Lindemann, Marie Bargiela, Marie Hofbauer,
 924 Omar Badri, et al. Cleanpatrick: A benchmark for image data cleaning. *arXiv preprint*
 925 *arXiv:2505.11034*, 2025.

926 Matthew Groh, Caleb Harris, Luis Soenksen, Felix Lau, Rachel Han, Aerin Kim, Arash Koochek,
 927 and Omar Badri. Evaluating deep neural networks trained on clinical images in dermatology with the
 928 fitzpatrick 17k dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern*
 929 *Recognition*, pp. 1820–1828, 2021.

930 Groq. Groqcloud supported models, 2025. URL <https://console.groq.com/docs/models>. Accessed: 2025-05-16.

931

932

933 Xuehang Guo, Xingyao Wang, Yangyi Chen, Sha Li, Chi Han, Manling Li, and Heng Ji. Sync-
 934 mind: Measuring agent out-of-sync recovery in collaborative software engineering. *arXiv preprint*
 935 *arXiv:2502.06994*, 2025.

936 Yaming Guo, Kai Guo, Xiaofeng Cao, Tieru Wu, and Yi Chang. Out-of-distribution generalization
 937 of federated learning via implicit invariant relationships. In *International Conference on Machine*
 938 *Learning*, pp. 11905–11933. PMLR, 2023.

939

940 Shyam Gupta. Fracatlas. <https://www.kaggle.com/datasets/shyamgupta196/fracatlas>, 2023. Accessed: 2025-05-22.

941

942 Izzeddin Gur, Hiroki Furuta, Austin V Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and
 943 Aleksandra Faust. A real-world webagent with planning, long context understanding, and program
 944 synthesis. In *The Twelfth International Conference on Learning Representations*.

945

946 David Gutman, Noel CF Codella, Emre Celebi, Brian Helba, Michael Marchetti, Nabin Mishra, and
 947 Allan Halpern. Skin lesion analysis toward melanoma detection: A challenge at the international
 948 symposium on biomedical imaging (isbi) 2016, hosted by the international skin imaging collaboration
 949 (isic). *arXiv preprint arXiv:1605.01397*, 2016.

950 Haashaatif. Melanoma histopathology dataset. <https://www.kaggle.com/datasets/haashaatif/melanoma-histopathology-dataset>, 2024. Accessed: 2024-05-22.

951

952 Senyu Han, Lu Chen, Li-Min Lin, Zhengshan Xu, and Kai Yu. Ibsen: Director-actor agent collab-
 953 oration for controllable and interactive drama script generation. *arXiv preprint arXiv:2407.01093*,
 954 2024a.

955

956 Shijie Han, Changhai Zhou, Yiqing Shen, Tianning Sun, Yuhua Zhou, Xiaoxia Wang, Zhixiao Yang,
 957 Jingshu Zhang, and Hongguang Li. Finsphere: A conversational stock analysis agent equipped with
 958 quantitative tools based on real-time database. *arXiv preprint arXiv:2501.12399*, 2025.

959

960 Xuewen Han, Neng Wang, Shangkun Che, Hongyang Yang, Kunpeng Zhang, and Sean Xin Xu. En-
 961 hancing investment analysis: Optimizing ai-agent collaboration in financial research. In *Proceedings*
 962 *of the 5th ACM International Conference on AI in Finance*, pp. 538–546, 2024b.

963 Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin
 964 Burns, Samir Puranik, Horace He, Dawn Song, and Jacob Steinhardt. Measuring Coding Challenge
 965 Competence With APPS, November 2021. URL <http://arxiv.org/abs/2105.09938>.
 966 *arXiv:2105.09938 [cs]*.

967 HFUTYBX. Mhsitcholedoch dataset preprocessed. <https://www.kaggle.com/datasets/hfutybx/mhsitcholedoch-dataset-preprocessed-dataset>, 2024. Accessed:
 968 2024-05-22.

969

970 Hmchuong. X-ray bone shadow suppression. <https://www.kaggle.com/datasets/hmchuong/xray-bone-shadow-suppression>, 2023. Accessed: 2025-05-22.

971

972 Huthayfa Hodeb. Nih chest x-rays (bbox version). <https://www.kaggle.com/datasets/huthayfahodeb/nih-chest-x-rays-bbox-version>, 2023. Accessed: 2025-05-22.

973

974

975 Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data
976 distribution for federated visual classification, 2019. URL <https://arxiv.org/abs/1909.06335>.

977

978 Ruida Hu, Chao Peng, Xinchen Wang, and Cuiyun Gao. An llm-based agent for reliable docker
979 environment configuration. *arXiv preprint arXiv:2502.13681*, 2025.

980

981 Dong Huang, Jie M. Zhang, Michael Luck, Qingwen Bu, Yuhao Qing, and Heming Cui. AgentCoder:
982 Multi-Agent-based Code Generation with Iterative Testing and Optimisation, May 2024a. URL
983 <http://arxiv.org/abs/2312.13010>. arXiv:2312.13010 [cs].

984

985 Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. MLAgentBench: Evaluating Language
986 Agents on Machine Learning Experimentation. In *Forty-first International Conference on Machine
987 Learning*, June 2024b. URL <https://openreview.net/forum?id=1Fs1LvjYQW>.

988

989 IMT Kaggle Team. Dental radiography dataset. <https://www.kaggle.com/datasets/imtkaggleteam/dental-radiography>, 2023. Accessed: 2025-05-22.

990

991 Yoshitaka Inoue, Tianci Song, and Tianfan Fu. Drugagent: Explainable drug repurposing agent with
992 large language model-based reasoning. *arXiv preprint arXiv:2408.13378*, 2024.

993

994 Md Ashraful Islam, Mohammed Eunus Ali, and Md Rizwan Parvez. Codesim: Multi-agent code
995 generation and problem solving through simulation-driven planning and debugging. *arXiv preprint
arXiv:2502.05664*, 2025.

996

997 Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando
998 Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and Contamination Free
999 Evaluation of Large Language Models for Code, June 2024. URL <http://arxiv.org/abs/2403.07974>. arXiv:2403.07974 [cs].

1000

1001 Naman Jain, Jaskirat Singh, Manish Shetty, Liang Zheng, Koushik Sen, and Ion Stoica. R2e-gym:
1002 Procedural environments and hybrid verifiers for scaling open-weights swe agents. *arXiv preprint
arXiv:2504.07164*, 2025.

1003

1004 Yixing Jiang, Kameron C Black, Gloria Geng, Danny Park, James Zou, Andrew Y Ng, and Jonathan H
1005 Chen. Medagentbench: A realistic virtual ehr environment to benchmark medical llm agents. *arXiv
1006 preprint arXiv:2501.14654*, 2025.

1007

1008 Carlos E. Jimenez, John Yang, Alexander Wetzig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik
1009 Narasimhan. SWE-bench: Can Language Models Resolve Real-World GitHub Issues?, April 2024.
1010 URL <http://arxiv.org/abs/2310.06770>. arXiv:2310.06770 [cs].

1011

1012 Liqiang Jing, Zhehui Huang, Xiaoyang Wang, Wenlin Yao, Wenhao Yu, Kaixin Ma, Hongming Zhang,
1013 Xinya Du, and Dong Yu. DSbench: How Far Are Data Science Agents to Becoming Data Science
1014 Experts?, September 2024. URL <http://arxiv.org/abs/2409.07703>. arXiv:2409.07703
[cs].

1015

1016 Lin Justin. Train val test tcga coad msi mss. <https://www.kaggle.com/datasets/linjustin/train-val-test-tcga-coad-msi-mss>, 2024. Accessed: 2024-05-22.

1017

1018 Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and
1019 Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In
1020 *International conference on machine learning*. PMLR, 2020.

1021

1022 Jeremy Kawahara, Sara Daneshvar, Giuseppe Argenziano, and Ghassan Hamarneh. Seven-point
1023 checklist and skin lesion classification using multitask multimodal neural nets. *IEEE journal of
1024 biomedical and health informatics*, 23(2):538–546, 2018.

1025

Yubin Kim, Chanwoo Park, Hyewon Jeong, Yik Siu Chan, Xuhai Xu, Daniel McDuff, Hyeonhoon
Lee, Marzyeh Ghassemi, Cynthia Breazeal, Hae Park, et al. Mdagents: An adaptive collaboration

1026 of llms for medical decision-making. *Advances in Neural Information Processing Systems*, 37: 1027 79410–79452, 2024.

1028

1029 Felipe Kitamura. Spr x-ray age and gender dataset. <https://www.kaggle.com/datasets/felipekitamura/spr-x-ray-age-and-gender-dataset>, 2022a. Accessed: 2025-05-22.

1030

1031

1032 Felipe Kitamura. Unifesp x-ray bodypart classification. <https://www.kaggle.com/datasets/felipekitamura/unifesp-xray-bodypart-classification>, 2022b. Accessed: 2025-05-22.

1033

1034

1035

1036 Sourabh Kumar. Breast cancer histopathology. <https://www.kaggle.com/datasets/sourabhkumar29/breast-cancer-histopathology>, 2024. Accessed: 2024-05-22.

1037

1038 Shrinidhi Kumbhar, Venkatesh Mishra, Kevin Coutinho, Divij Handa, Ashif Iquebal, and Chitta Baral.

1039 Hypothesis generation for materials discovery and design using goal-driven and constraint-guided

1040 llm agents. *arXiv preprint arXiv:2501.13299*, 2025.

1041

1042 The Red Lad. Pannuke dataset (experimental). <https://www.kaggle.com/datasets/theredlad/pannuke-dataset-experimental-data>, 2024. Accessed: 2024-05-22.

1043

1044

1045 Pamela J LaMontagne, Tammie LS Benzinger, John C Morris, Sarah Keefe, Russ Hornbeck, Chengjie

1046 Xiong, Elizabeth Grant, Jason Hassenstab, Krista Moulder, Andrei G Vlassenko, et al. OASIS-3:

1047 longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and Alzheimer disease.

1048 *MedRxiv*, pp. 1–37, 2019.

1049

1050 Langgraph. Langgraph documentation, 2025. URL <https://www.langchain.com/langgraph>. Accessed: 2025-05-16.

1051

1052 Gisang Lee, Sangwoo Park, Junyoung Park, Andrew Chung, Sieun Park, Yoonah Park, Byungju Kim,

1053 and Min-gyu Cho. Expanding search space with diverse prompting agents: An efficient sampling

1054 approach for llm mathematical reasoning. *arXiv preprint arXiv:2410.09780*, 2024.

1055

1056 Jingoo Lee, Kyungho Lim, Young-Chul Jung, and Byung-Hoon Kim. Psyche: A multi-faceted

1057 patient simulation framework for evaluation of psychiatric assessment conversational agents. *arXiv*

1058 *preprint arXiv:2501.01594*, 2025.

1059

1060 Tahir Lee. Uterine leiomyosarcoma histopathology. <https://www.kaggle.com/datasets/tahirlee/uterine-leiomyosarcoma-histopathology>, 2024. Accessed: 2024-05-22.

1061

1062 Bin Lei, Yi Zhang, Shan Zuo, Ali Payani, and Caiwen Ding. Macm: Utilizing a multi-agent system

1063 for condition mining in solving complex mathematical problems. *arXiv preprint arXiv:2404.04735*,

1064 2024.

1065

1066 Bin Xu Li, Tiankai Yan, Yuanting Pan, Jie Luo, Ruiyang Ji, Jiayuan Ding, Zhe Xu, Shilong Liu, Haoyu

1067 Dong, Zihao Lin, et al. Mmedagent: Learning to use medical tools with multi-modal agent. In

1068 *EMNLP (Findings)*, 2024a.

1069

1070 Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation, 2019.

1071 URL <https://arxiv.org/abs/1910.03581>.

1072

1073 Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbulin, and Bernard Ghanem. Camel:

1074 Communicative agents for "mind" exploration of large language model society. *Advances in Neural*

1075 *Information Processing Systems*, 36:51991–52008, 2023a.

1076

1077 Long Li, Weiwen Xu, Jiayan Guo, Ruochen Zhao, Xingxuan Li, Yuqian Yuan, Boqiang Zhang,

1078 Yuming Jiang, Yifei Xin, Ronghao Dang, et al. Chain of ideas: Revolutionizing research via novel

1079 idea development with llm agents. *arXiv preprint arXiv:2410.13185*, 2024b.

1080

1081 Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning, 2021a. URL

1082 <https://arxiv.org/abs/2103.16257>.

1080 Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. A
 1081 survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE*
 1082 *Transactions on Knowledge and Data Engineering*, 35(4):3347–3366, 2021b.

1083

1084 Qinbin Li, Bingsheng He, and Dawn Song. Adversarial collaborative learning on non-IID features. In
 1085 Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan
 1086 Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202
 1087 of *Proceedings of Machine Learning Research*, pp. 19504–19526. PMLR, 23–29 Jul 2023b. URL
 1088 <https://proceedings.mlr.press/v202/li23j.html>.

1089

1090 Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges,
 1091 methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020a.

1092

1093 Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith.
 1094 Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:
 1095 429–450, 2020b.

1096

1097 Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated
 1098 learning through personalization, 2021c. URL <https://arxiv.org/abs/2012.04221>.

1099

1100 Vincent Li, Yule Fu, Tim Knappe, Kevin Han, and Kevin Zhu. Automating mathematical proof gen-
 1101 eration using large language model agents and knowledge graphs. *arXiv preprint arXiv:2503.11657*,
 1102 2025.

1103

1104 Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on
 1105 non-iid features via local batch normalization, 2021d. URL <https://arxiv.org/abs/2102.07623>.

1106

1107 Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B. Allen, Randy P. Auerbach, David Brent, Ruslan
 1108 Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with
 1109 local and global representations, 2020. URL <https://arxiv.org/abs/2001.01523>.

1110

1111 Xun Liang, Jiawei Yang, Yezhaozhi Wang, Chen Tang, Zifan Zheng, Simin Niu, Shichao Song,
 1112 Hanyu Wang, Bo Tang, Feiyu Xiong, et al. Surveyx: Academic survey automation via large language
 1113 models. *arXiv preprint arXiv:2502.14776*, 2025.

1114

1115 Sook-Lei Liew, Bethany P Lo, Miranda R Donnelly, Artemis Zavaliangos-Petropulu, Jessica N
 1116 Jeong, Giuseppe Barisano, Alexandre Hutton, Julia P Simon, Julia M Juliano, Anisha Suri, et al. A
 1117 large, curated, open-source stroke neuroimaging dataset to improve lesion segmentation algorithms.
 1118 *Scientific data*, 9(1):320, 2022.

1119

1120 Ryan Y Lin, Siddhartha Ojha, Kevin Cai, and Maxwell F Chen. Strategic collusion of llm agents:
 1121 Market division in multi-commodity competitions. *arXiv preprint arXiv:2410.00031*, 2024.

1122

1123 Ben Liu, Jihan Zhang, Fangquan Lin, Xu Jia, and Min Peng. One size doesn't fit all: A personalized
 1124 conversational tutoring agent for mathematics instruction. 2025. URL <https://arxiv.org/abs/2502.12633>.

1125

1126 Hong-Hsiang Liu and Yi-Wen Liu. Agent-driven large language models for mandarin lyric generation.
 1127 In *2024 27th Conference of the Oriental COCOSDA International Committee for the Co-ordination
 1128 and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA)*, pp. 1–6. IEEE,
 1129 2024.

1130

1131 Renpu Liu, Cong Shen, and Jing Yang. Federated representation learning in the under-parameterized
 1132 regime, 2024a. URL <https://arxiv.org/abs/2406.04596>.

1133

1134 Siyi Liu, Chen Gao, and Yong Li. Large Language Model Agent for Hyper-Parameter Optimization,
 1135 February 2024b. URL <http://arxiv.org/abs/2402.01881>. arXiv:2402.01881 [cs].

1136

1137 Lokisilvres. Dental disease panoramic detection dataset. <https://www.kaggle.com/datasets/lokisilvres/dental-disease-panoramic-detection-dataset>,
 1138 2023. Accessed: 2025-05-22.

1134 Wang Lu, Jindong Wang, Yiqiang Chen, Xin Qin, Renjun Xu, Dimitrios Dimitriadis, and Tao
 1135 Qin. Personalized federated learning with adaptive batchnorm for healthcare, 2022. URL <https://arxiv.org/abs/2112.00734>.
 1136

1137 Yuxuan Lu, Bingsheng Yao, Hansu Gu, Jing Huang, Jessie Wang, Laurence Li, Jiri Gesi, Qi He, Toby
 1138 Jia-Jun Li, and Dakuo Wang. Uxagent: An llm agent-based usability testing framework for web
 1139 design. *arXiv preprint arXiv:2502.12561*, 2025.
 1140

1141 Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi
 1142 Chen, Ziyue Qiao, Qingqing Long, et al. Large language model agent: A survey on methodology,
 1143 applications and challenges. *arXiv preprint arXiv:2503.21460*, 2025.
 1144

1145 Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. No fear of heterogeneity:
 1146 Classifier calibration for federated learning with non-iid data, 2021. URL <https://arxiv.org/abs/2106.05001>.
 1147

1148 Tianmi Ma, Jiawei Du, Wenxin Huang, Wenjie Wang, Liang Xie, Xian Zhong, and Joey Tianyi Zhou.
 1149 Llm knows geometry better than algebra: Numerical understanding of llm-based agents in a trading
 1150 arena. *arXiv preprint arXiv:2502.17967*, 2025.
 1151

1152 Xiaosong Ma, Jie Zhang, Song Guo, and Wencho Xu. Layer-wised model aggregation for person-
 1153 alized federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and*
 1154 *Pattern Recognition (CVPR)*, pp. 10092–10101, June 2022.
 1155

1156 Nikita Manaenkov. Annotated x-ray angiography dataset. <https://www.kaggle.com/datasets/nikitamanaenkov/annotated-x-ray-angiography-dataset>, 2023.
 1157 Accessed: 2025-05-22.
 1158

1159 Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. Federated
 1160 multi-task learning under a mixture of distributions, 2022. URL <https://arxiv.org/abs/2108.10252>.
 1161

1162 Antonis Maronikolakis, Ana Peleteiro Ramallo, Weiwei Cheng, and Thomas Kober. What should i
 1163 wear to a party in a greek taverna? evaluation for conversational agents in the fashion domain. *arXiv*
 1164 *preprint arXiv:2408.08907*, 2024.
 1165

1166 Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas.
 1167 Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence*
 1168 *and statistics*. PMLR, 2017.
 1169

1170 Kai Mei, Xi Zhu, Wujiang Xu, Wenyue Hua, Mingyu Jin, Zelong Li, Shuyuan Xu, Ruosong
 1171 Ye, Yingqiang Ge, and Yongfeng Zhang. Aios: Llm agent operating system. *arXiv preprint*
 1172 *arXiv:2403.16971*, 2024.
 1173

1174 Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin
 1175 Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. The multimodal brain
 1176 tumor image segmentation benchmark (BRATS). *IEEE Transactions on Medical Imaging*, 34(10):
 1993–2024, 2014.
 1177

1178 Arindam Mitra, Luciano Del Corro, Guoqing Zheng, Shweta Mahajan, Dany Rouhana, Andres Codas,
 1179 Yadong Lu, Wei-ge Chen, Olga Vrousgos, Corby Rosset, et al. Agentinstruct: Toward generative
 1180 teaching with agentic flows. *arXiv preprint arXiv:2407.03502*, 2024.
 1181

1182 Paul Timothy Mooney. Chest x-ray pneumonia. <https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia>, 2018. Accessed: 2025-05-22.
 1183

1184 Paul Timothy Mooney. Breast histopathology images. <https://www.kaggle.com/datasets/paultimothymooney/breast-histopathology-images>, 2024. Ac-
 1185 cessed: 2024-05-22.
 1186

1187 WSH Munirah. Npc-88k-public dataset. <https://www.kaggle.com/datasets/wshmunirah/npc-88k-public>, 2024. Accessed: 2024-05-22.
 1188

1188 Andrew MVD. Lung and colon cancer histopathological images. <https://www.kaggle.com/datasets/andrewmvd/lung-and-colon-cancer-histopathological-images>, 1189 2024a. Accessed: 2024-05-22.

1190

1191 Andrew MVD. Breast cancer cell segmentation. <https://www.kaggle.com/datasets/andrewmvd/breast-cancer-cell-segmentation>, 2024b. Accessed: 2024-05-22.

1192

1193

1194 Mouheb Ben Nasr and Yassine Hachaïchi. Reinforcement learning agent for client selection in 1195 federated llms.

1196

1197 A. Tuan Nguyen, Philip Torr, and Ser Nam Lim. Fedsr: A simple and effec- 1198 tive domain generalization method for federated learning. In S. Koyejo, S. Mo- 1199 hamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural In- 1200 formation Processing Systems*, volume 35, pp. 38831–38843. Curran Associates, Inc., 1201 2022a. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/fd946a6c99541fddc3d64a3ea39a1bc2-Paper-Conference.pdf.

1202

1203 Dinh C Nguyen, Quoc-Viet Pham, Pubudu N Pathirana, Ming Ding, Aruna Seneviratne, Zihuai Lin, 1204 Octavia Dobre, and Won-Joo Hwang. Federated learning for smart healthcare: A survey. *ACM 1205 Computing Surveys (CSUR)*, 55(3):1–37, 2022b.

1206

1207 Huan Ning, Zhenlong Li, Temitope Akinboyewa, and M Naser Lessani. An autonomous gis agent 1208 framework for geospatial data retrieval. *International Journal of Digital Earth*, 18(1):2458688, 2025.

1209

1210 Jaehoon Oh, Sangmook Kim, and Se-Young Yun. Fedbabu: Towards enhanced representation for 1211 federated image classification, 2022. URL <https://arxiv.org/abs/2106.06042>.

1212

1213 Izunna Okpala, Ashkan Golgoon, and Arjun Ravi Kannan. Agentic ai systems applied to tasks in 1214 financial services: Modeling and model risk management crews. *arXiv preprint arXiv:2502.05439*, 1215 2025.

1216

1217 OpenAI. Openai api documentation, 2025. URL <https://openai.com/api/>. Accessed: 1218 2025-05-16.

1219

1220 Orvile. Bone fracture dataset. <https://www.kaggle.com/datasets/orvile/bone-fracture-dataset>, 2023a. Accessed: 2025-05-22.

1221

1222 Orvile. Simple vs comminuted fractures x-ray data. <https://www.kaggle.com/datasets/orvile/simple-vs-comminuted-fractures-x-ray-data>, 2023b. Accessed: 2025-05-22.

1223

1224 Orvile. Human bone fractures image dataset (hbfmid). <https://www.kaggle.com/datasets/orvile/human-bone-fractures-image-dataset-hbfmid>, 2023c. Ac- 1225 cessed: 2025-05-22.

1226

1227 Orvile. Digital knee x-ray images. <https://www.kaggle.com/datasets/orvile/digital-knee-x-ray-images>, 2023d. Accessed: 2025-05-22.

1228

1229 Orvile. Gastric cancer histopathology tissue image 1230 dataset. <https://www.kaggle.com/datasets/orvile/gastric-cancer-histopathology-tissue-image-dataset>, 2024. Accessed: 1231 2024-05-22.

1232

1233 Andre GC Pacheco, Gustavo R Lima, Amanda S Salomão, Breno Krohling, Igor P Biral, Gabriel G 1234 de Angelo, Fábio CR Alves Jr, José GM Esgario, Alana C Simora, Pedro BC Castro, et al. PAD-UFES- 1235 20: A skin lesion dataset composed of patient data and clinical images collected from smartphones. 1236 *Data in Brief*, 32:1–10, 2020.

1237

1238 Jiayi Pan, Xingyao Wang, Graham Neubig, Navdeep Jaithly, Heng Ji, Alane Suhr, and Yizhe Zhang. 1239 Training software engineering agents and verifiers with swe-gym. *arXiv preprint arXiv:2412.21139*, 2024.

1240

1241 Emilio Pasquel. Deep learning driven diagnosis of humerus fractures. <https://www.kaggle.com/datasets/emiliopaspuel/>

1242 deeplearning-driven-diagnosis-of-humerus-fractures, 2024. Accessed:
1243 2025-05-22.

1244 Bjarne Pfitzner, Nico Steckhan, and Bert Arnrich. Federated learning in a medical context: A
1245 systematic literature review. *ACM Transactions on Internet Technology (TOIT)*, 21(2):1–31, 2021.

1246 Bits N Pieces. Ovarian cancer and subtypes dataset histopathology. <https://www.kaggle.com/datasets/bitsnipes/ovarian-cancer-and-subtypes-dataset-histopathology>, 2024. Accessed:
1247 2024-05-22.

1248 Jianing Qiu, Kyle Lam, Guohao Li, Amish Acharya, Tien Yin Wong, Ara Darzi, Wu Yuan, and Eric J
1249 Topol. Llm-based agentic systems in medicine and healthcare. *Nature Machine Intelligence*, 6(12):
1250 1418–1420, 2024.

1251 Xiangyan Qu, Gaopeng Gou, Jiamin Zhuang, Jing Yu, Kun Song, Qihao Wang, Yili Li, and Gang
1252 Xiong. Proapo: Progressively automatic prompt optimization for visual classification. In *Proceedings
1253 of the Computer Vision and Pattern Recognition Conference*, pp. 25145–25155, 2025.

1254 Tawsifur Rahman. Covid-19 radiography database. <https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database>, 2020. Accessed: 2025-05-22.

1255 Tawsifur Rahman. Aseptic loose hip implant x-ray database. <https://www.kaggle.com/datasets/tawsifurrahman/aseptic-loose-hip-implant-xray-database>,
1256 2022. Accessed: 2025-05-22.

1257 Suraj Rajendran, Jihad S Obeid, Hamidullah Binol, Kristie Foley, Wei Zhang, Philip Austin, Joey
1258 Brakefield, Metin N Gurcan, and Umit Topaloglu. Cloud-based federated learning implementation
1259 across medical centers. *JCO Clinical Cancer Informatics*, 5:1–11, 2021.

1260 Kiran Ramnath, Kang Zhou, Sheng Guan, Soumya Smruti Mishra, Xuan Qi, Zhengyuan Shen, Shuai
1261 Wang, Sangmin Woo, Sullam Jeoung, Yawei Wang, et al. A systematic survey of automatic prompt
1262 optimization techniques. *arXiv preprint arXiv:2502.16923*, 2025.

1263 RANZCR. Ranzcr clip - catheter and line classification. <https://www.kaggle.com/c/ranzcr-clip-catheter-line-classification>, 2021. Accessed: 2025-05-22.

1264 Reasat. Histo image-text. <https://www.kaggle.com/datasets/reasat/histo-img-text>, 2024. Accessed: 2024-05-22.

1265 Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný,
1266 Sanjiv Kumar, and H. Brendan McMahan. Adaptive federated optimization, 2021. URL <https://arxiv.org/abs/2003.00295>.

1267 Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon
1268 Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. The future of digital health
1269 with federated learning. *NPJ digital medicine*, 3(1):119, 2020.

1270 B Madushani Rodrigo. Fracture multi-region x-ray data. <https://www.kaggle.com/datasets/bmadushanirodrigo/fracture-multi-region-x-ray-data>, 2022.
1271 Accessed: 2025-05-22.

1272 Daniel Rose, Chia-Chien Hung, Marco Lepri, Israa Alqassem, Kiril Gashtelovski, and Carolin
1273 Lawrence. Meddxagent: A unified modular agent framework for explainable automatic differential
1274 diagnosis. *arXiv preprint arXiv:2502.19175*, 2025.

1275 Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-
1276 agnostic distributed multi-task optimization under privacy constraints, 2019. URL <https://arxiv.org/abs/1910.01991>.

1277 Samuel Schmidgall and Michael Moor. Agentrxiv: Towards collaborative autonomous research.
1278 *arXiv preprint arXiv:2503.18102*, 2025.

1296 Dominik Schmidt, Zhengyao Jiang, and Yuxiang Wu. Introducing Weco AIDE, April 2024. URL
 1297 <https://www.weco.ai/blog/technical-report>.
 1298

1299 Jonathan Scott, Hossein Zakerinia, and Christoph H. Lampert. Pefl: Personalized federated learning
 1300 by learning to learn, 2025. URL <https://arxiv.org/abs/2306.05515>.
 1301

1302 Constantin Seibold. Anatomy in chest x-rays (pax-ray++). <https://www.kaggle.com/datasets/constantinseibold/anatomy-in-chest-x-rays-pax-ray>, 2023. Ac-
 1303 cessed: 2025-05-22.
 1304

1305 Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using
 1306 hypernetworks, 2021. URL <https://arxiv.org/abs/2103.04628>.
 1307

1308 Zitong Shi, Guancheng Wan, Wenke Huang, Guibin Zhang, Jiawei Shao, Mang Ye, and Carl Yang.
 1309 Privacy-enhancing paradigms within federated multi-agent systems. In *ICML 2025 Workshop on*
 1310 *Collaborative and Federated Agentic Workflows*.
 1311

1312 Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. Agentic retrieval-augmented
 1313 generation: A survey on agentic rag. *arXiv preprint arXiv:2501.09136*, 2025.
 1314

1315 Jayanthi Sivaswamy, SR Krishnadas, Gopal Datt Joshi, Madhulika Jain, and A Ujjwaal Syed Tabish.
 1316 Drishti-gs: Retinal image dataset for optic nerve head (ohn) segmentation. In *2014 IEEE 11th*
 1317 *international symposium on biomedical imaging (ISBI)*, pp. 53–56. IEEE, 2014.
 1318

1319 Spritan1. Yolo annotated chestxray 8 object detection. <https://www.kaggle.com/datasets/spritan1/yolo-annotated-chestxray-8-object-detection>, 2023.
 1320 Accessed: 2025-05-22.
 1321

1322 Ian Steenstra, Farnaz Nouraei, and Timothy W Bickmore. Scaffolding empathy: Training coun-
 1323 selors with simulated patients and utterance-level performance visualizations. *arXiv preprint*
 1324 *arXiv:2502.18673*, 2025.
 1325

1326 Osama H Taher. Heel dataset. <https://www.kaggle.com/datasets/osamahtaher/heel-dataset>, 2023. Accessed: 2025-05-22.
 1327

1328 Jiahao Tan and Xinpeng Wang. FL-bench: A federated learning benchmark for solving image
 1329 classification tasks. URL <https://github.com/KarhouTam/FL-bench>.
 1330

1331 Jiahao Tan, Yipeng Zhou, Gang Liu, Jessie Hui Wang, and Shui Yu. pfedsim: Similarity-aware
 1332 model aggregation towards personalized federated learning. *arXiv preprint arXiv:2305.15706*, 2023.
 1333

1334 Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto:
 1335 Federated prototype learning across heterogeneous clients, 2022. URL <https://arxiv.org/abs/2105.00243>.
 1336

1337 Xiangru Tang, Yuliang Liu, Zefan Cai, Yanjun Shao, Junjie Lu, Yichi Zhang, Zexuan Deng, Helan
 1338 Hu, Kaikai An, Ruijun Huang, Shuzheng Si, Sheng Chen, Haozhe Zhao, Liang Chen, Yan Wang,
 1339 Tianyu Liu, Zhiwei Jiang, Baobao Chang, Yin Fang, Yujia Qin, Wangchunshu Zhou, Yilun Zhao,
 1340 Arman Cohan, and Mark Gerstein. ML-Bench: Evaluating Large Language Models and Agents for
 1341 Machine Learning Tasks on Repository-Level Code, August 2024. URL <http://arxiv.org/abs/2311.09835>.
 1342

1343 Xiangru Tang, Tianyu Hu, Muyang Ye, Yanjun Shao, Xunjian Yin, Siru Ouyang, Wangchunshu Zhou,
 1344 Pan Lu, Zhuseng Zhang, Yilun Zhao, et al. Chemagent: Self-updating library in large language
 1345 models improves chemical reasoning. *arXiv preprint arXiv:2501.06590*, 2025.
 1346

1347 Tapendu. Chest x-ray dataset for tuberculosis segmentation. <https://www.kaggle.com/datasets/iamtapendu/chest-x-ray-lungs-segmentation>, 2023a. Accessed:
 1348 2025-05-22.
 1349

1349 Tapendu. Rsna pneumonia processed dataset. <https://www.kaggle.com/datasets/iamtapendu/rsna-pneumonia-processed-dataset>, 2023b. Accessed: 2025-05-22.

1350 Mr. Tejas. Covid-19 and normal x-ray dataset (balanced). <https://www.kaggle.com/datasets/mrtejas/covid-19-and-normal-x-ray-dataset-balanced>, 2022.
 1351 Accessed: 2025-05-22.

1352

1353 Aryamaan Thakur. Rsna breast cancer detection roi 1024. <https://www.kaggle.com/datasets/aryamaanthakur/rsna-breast-cancer-detection-roi-1024>, 2024.
 1354 Accessed: 2025-05-22.

1355

1356

1357 TrainingDataPro. Chest x-ray 17 diseases. <https://www.kaggle.com/datasets/trainingdatapro/chest-xray-17-diseases>, 2023. Accessed: 2025-05-22.

1358

1359

1360 Prashant Trivedi, Souradip Chakraborty, Avinash Reddy, Vaneet Aggarwal, Amrit Singh Bedi, and George K Atia. Align-pro: A principled approach to prompt optimization for llm alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 27653–27661, 2025.

1361

1362

1363 TruthIsNeverLinear. Brecahad. <https://www.kaggle.com/datasets/truthisneverlinear/brecahad>, 2024. Accessed: 2024-05-22.

1364

1365

1366 Preet Viradiya. Covid-19 radiography dataset extended. <https://www.kaggle.com/datasets/preetviradiya/covid19-radiography-dataset>, 2023. Accessed: 2025-05-22.

1367

1368

1369 Xiangpeng Wan, Haicheng Deng, Kai Zou, and Shiqi Xu. Enhancing the efficiency and accuracy of underlying asset reviews in structured finance: The application of multi-agent framework. *arXiv preprint arXiv:2405.04294*, 2024.

1370

1371

1372 Cunshi Wang, Xinjie Hu, Yu Zhang, Xunhao Chen, Pengliang Du, Yiming Mao, Rui Wang, Yuyang Li, Ying Wu, Hang Yang, et al. Starwhisper telescope: Agent-based observation assistant system to approach ai astrophysicist. *arXiv preprint arXiv:2412.06412*, 2024a.

1373

1374

1375

1376 Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

1377

1378

1379 Jian Wang, Yinpei Dai, Yichi Zhang, Ziqiao Ma, Wenjie Li, and Joyce Chai. Training turn-by-turn verifiers for dialogue tutoring agents: The curious case of llms as your coding tutors. *arXiv preprint arXiv:2502.13311*, 2025a.

1380

1381

1382

1383 Jiuniu Wang, Zehua Du, Yuyuan Zhao, Bo Yuan, Kexiang Wang, Jian Liang, Yaxi Zhao, Yihen Lu, Gengliang Li, Junlong Gao, et al. Aesopagent: Agent-driven evolutionary system on story-to-video production. *arXiv preprint arXiv:2403.07952*, 2024b.

1384

1385

1386 Ruida Wang, Rui Pan, Yuxin Li, Jipeng Zhang, Yizhen Jia, Shizhe Diao, Renjie Pi, Junjie Hu, and Tong Zhang. Ma-lot: Multi-agent lean-based long chain-of-thought reasoning enhances formal theorem proving. *arXiv preprint arXiv:2503.03205*, 2025b.

1387

1388

1389

1390 Xin Wang, Yifan Zhang, Xiaojing Zhang, Longhui Yu, Xinnna Lin, Jindong Jiang, Bin Ma, and Kaicheng Yu. Patentagent: Intelligent agent for automated pharmaceutical patent analysis. *arXiv preprint arXiv:2410.21312*, 2024c.

1391

1392

1393 Zhizheng Wang, Qiao Jin, Chih-Hsuan Wei, Shubo Tian, Po-Ting Lai, Qingqing Zhu, Chi-Ping Day, Christina Ross, and Zhiyong Lu. Geneagent: self-verification language agent for gene set knowledge discovery using domain databases. *arXiv preprint arXiv:2405.16205*, 2024d.

1394

1395

1396

1397 Ziyue Wang, Junde Wu, Chang Han Low, and Yueming Jin. Medagent-pro: Towards multi-modal evidence-based medical diagnosis via reasoning agentic workflow. *arXiv preprint arXiv:2503.18968*, 2025c.

1398

1399

1400 Abbi Ward, Jimmy Li, Julie Wang, Sriram Lakshminarasimhan, Ashley Carrick, Bilson Campana, Jay Hartford, Pradeep K. Sreenivasiah, Tiya Tiyasirisokchai, Sunny Virmani, Renee Wong, Yossi Matias, Greg S. Corrado, Dale R. Webster, Margaret Ann Smith, Dawn Siegel, Steven Lin, Justin Ko, Alan Karthikesalingam, Christopher Semturs, and Pooja Rao. Creating an empirical dermatology dataset through crowdsourcing with web search advertisements. *JAMA Network Open*, 7(11):

1401

1402

1403

1404 e2446615–e2446615, 11 2024. ISSN 2574-3805. 10.1001/jamanetworkopen.2024.46615. URL
 1405 <https://doi.org/10.1001/jamanetworkopen.2024.46615>.

1406

1407 Robert Wasenmüller, Kevin Hilbert, and Christoph Benzmüller. Script-based dialog policy planning
 1408 for llm-powered conversational agents: A basic architecture for an "ai therapist". *arXiv preprint*
 1409 *arXiv:2412.15242*, 2024.

1410 Panlong Wu, Kangshuo Li, Junbao Nan, and Fangxin Wang. Federated in-context llm agent learning.
 1411 *arXiv preprint arXiv:2412.08054*, 2024.

1412

1413 Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun
 1414 Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent
 1415 conversation. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

1416 Wenbei Xie, Donglin Liu, Haoran Yan, Wenjie Wu, and Zongyang Liu. Mathlearner: A large
 1417 language model agent framework for learning to solve mathematical problems. *arXiv preprint*
 1418 *arXiv:2408.01779*, 2024.

1419

1420 Ancheng Xu, Di Yang, Renhao Li, Jingwei Zhu, Minghuan Tan, Min Yang, Wanxin Qiu, Mingchen
 1421 Ma, Haihong Wu, Bingyu Li, et al. Autocbt: An autonomous multi-agent framework for cognitive
 1422 behavioral therapy in psychological counseling. *arXiv preprint arXiv:2501.09426*, 2025a.

1423 Zhenran Xu, Longyue Wang, Jifang Wang, Zhouyi Li, Senbao Shi, Xue Yang, Yiyu Wang, Baotian
 1424 Hu, Jun Yu, and Min Zhang. Filmagent: A multi-agent framework for end-to-end film automation in
 1425 virtual 3d spaces. *arXiv preprint arXiv:2501.12909*, 2025b.

1426

1427 Jian Yang, Wei Zhang, Jiaxi Yang, Yibo Miao, Shanghaoran Quan, Zhenhe Wu, Qiyo Peng, Liqun
 1428 Yang, Tianyu Liu, Zeyu Cui, et al. Multi-agent collaboration for multilingual code instruction tuning.
 1429 *arXiv preprint arXiv:2502.07487*, 2025a.

1430 Xiyuan Yang, Wenke Huang, and Mang Ye. Fedas: Bridging inconsistency in personalized federated
 1431 learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*
 1432 (CVPR), pp. 11986–11995, June 2024.

1433

1434 Yizhe Yang, Palakorn Achananuparp, Heyan Huang, Jing Jiang, Kit Phey Leng, Nicholas Gabriel
 1435 Lim, Cameron Tan Shi Ern, and Ee-peng Lim. Cami: A counselor agent supporting motivational
 1436 interviewing through state inference and topic exploration. *arXiv preprint arXiv:2502.02807*, 2025b.

1437 Yuzhe Yang, Yifei Zhang, Minghao Wu, Kaidi Zhang, Yunmiao Zhang, Honghai Yu, Yan Hu, and
 1438 Benyou Wang. Twinmarket: A scalable behavioral and socialsimulation for financial markets. *arXiv*
 1439 *preprint arXiv:2502.01506*, 2025c.

1440

1441 Zhiqin Yang, Yonggang Zhang, Yu Zheng, Xinmei Tian, Hao Peng, Tongliang Liu, and Bo Han.
 1442 Fedfed: Feature distillation against data heterogeneity in federated learning, 2023. URL <https://arxiv.org/abs/2310.05077>.

1443

1444 Beibei Yu, Tao Shen, Hongbin Na, Ling Chen, and Denqi Li. Mineagent: Towards remote-sensing
 1445 mineral exploration with multimodal large language models. *arXiv preprint arXiv:2412.17339*,
 1446 2024a.

1447

1448 Dingyao Yu, Kaitao Song, Peiling Lu, Tianyu He, Xu Tan, Wei Ye, Shikun Zhang, and Jiang Bian.
 1449 Musicagent: An ai agent for music understanding and generation with large language models. *arXiv*
 1450 *preprint arXiv:2310.11954*, 2023.

1451 Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng, Yuechen Jiang, Yupeng Cao, Zhi Chen,
 1452 Jordan Suchow, Zhenyu Cui, Rong Liu, et al. Fincon: A synthesized llm multi-agent system with
 1453 conceptual verbal reinforcement for enhanced financial decision making. *Advances in Neural*
 1454 *Information Processing Systems*, 37:137010–137045, 2024b.

1455

1456 Murong Yue, Wenhan Lyu, Wijdane Mifdal, Jennifer Suh, Yixuan Zhang, and Ziyu Yao. Mathvc:
 1457 An llm-simulated multi-character virtual classroom for mathematics education. *arXiv preprint*
 1458 *arXiv:2404.06711*, 2024.

1458 Taedong Yun, Eric Yang, Mustafa Safdari, Jong Ha Lee, Vaishnavi Vinod Kumar, S Sara Mahdavi,
 1459 Jonathan Amar, Derek Peyton, Reut Aharony, Andreas Michaelides, et al. Sleepless nights, sugary
 1460 days: Creating synthetic users with health conditions for realistic coaching agent interactions. *arXiv*
 1461 *preprint arXiv:2502.13135*, 2025.

1462 Jingying Zeng, Hui Liu, Zhenwei Dai, Xianfeng Tang, Chen Luo, Samarth Varshney, Zhen Li, and
 1463 Qi He. Cite before you speak: Enhancing context-response grounding in e-commerce conversational
 1464 llm-agents. *arXiv preprint arXiv:2503.04830*, 2025.

1465 Huan Zhang, Yu Song, Ziyu Hou, Santiago Miret, and Bang Liu. Honeycomb: A flexible llm-based
 1466 agent system for materials science. *arXiv preprint arXiv:2409.00135*, 2024a.

1467 Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan.
 1468 Fedala: Adaptive local aggregation for personalized federated learning. *Proceedings of the*
 1469 *AAAI Conference on Artificial Intelligence*, 37(9):11237–11244, June 2023. ISSN 2159-5399.
 1470 10.1609/aaai.v37i9.26330. URL <http://dx.doi.org/10.1609/aaai.v37i9.26330>.

1471 Jie Zhang, Zhiqi Li, Bo Li, Jianghe Xu, Shuang Wu, Shouhong Ding, and Chao Wu. Federated
 1472 learning with label distribution skew via logits calibration, 2022. URL <https://arxiv.org/abs/2209.00189>.

1473 Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai
 1474 Tian, Guoli Jia, Pengfei Li, et al. A survey of reinforcement learning for large reasoning models.
 1475 *arXiv preprint arXiv:2509.08827*, 2025.

1476 Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M. Alvarez. Personalized
 1477 federated learning with first order model optimization, 2021. URL <https://arxiv.org/abs/2012.08565>.

1478 Ran Zhang and Steffen Eger. Llm-based multi-agent poetry generation in non-cooperative environments.
 1479 *arXiv preprint arXiv:2409.03659*, 2024.

1480 Yiqun Zhang, Xiaocui Yang, Xiaobai Li, Siyuan Yu, Yi Luan, Shi Feng, Daling Wang, and Yifei
 1481 Zhang. Psydraw: A multi-agent multimodal system for mental health screening in left-behind
 1482 children. *arXiv preprint arXiv:2412.14769*, 2024b.

1483 Yuntong Zhang, Haifeng Ruan, Zhiyu Fan, and Abhik Roychoudhury. AutoCodeRover: Au-
 1484 tonomous Program Improvement, July 2024c. URL <http://arxiv.org/abs/2404.05427>.
 1485 arXiv:2404.05427 [cs].

1486 Mingkai Zheng, Xiu Su, Shan You, Fei Wang, Chen Qian, Chang Xu, and Samuel Albanie. Can
 1487 GPT-4 Perform Neural Architecture Search?, August 2023. URL <http://arxiv.org/abs/2304.10970>. arXiv:2304.10970 [cs].

1488 Yuan Zhou, Peng Zhang, Mengya Song, Alice Zheng, Yiwen Lu, Zhiheng Liu, Yong Chen, and
 1489 Zhaohan Xi. Zodiac: A cardiologist-level llm framework for multi-agent diagnostics. *arXiv preprint*
 1490 *arXiv:2410.02026*, 2024.

1491 Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous
 1492 federated learning, 2021. URL <https://arxiv.org/abs/2105.10056>.

1493

1494

1495

1496

1497

1498

1499

1500

1501

1502

1503

1504

1505

1506

1507

1508

1509

1510

1511

1512	APPENDIX	
1513		
1514		
1515	CONTENTS	
1516		
1517	1 Introduction and Background	1
1518		
1519	2 FedAgentBench Framework	3
1520	2.1 Problem Formulation and Overview	3
1521	2.2 Client Dataset Curation and FL Algorithm Integration	4
1522	2.3 Federated Agentic Framework Construction	5
1523	2.4 Privacy Preserving and Modular Design	6
1524		
1525		
1526		
1527	3 Experiments and Results	7
1528	3.1 Implementation and Evaluation Details	7
1529	3.2 Main Results and Key Insights	8
1530	3.3 RQ4: Agent Failure Analysis:	9
1531	3.4 Final Federated Training Performance:	10
1532		
1533		
1534		
1535	4 Conclusion and Limitation	10
1536		
1537		
1538	CONTENTS OF APPENDIX	29
1539		
1540	A Related Works	31
1541	A.1 Federated Learning for Medical Image Analysis	31
1542	A.2 LLM Agent Applications	31
1543	A.3 LLM Agents for Machine Learning, Software Engineering, and Federated Learning	32
1544		
1545		
1546		
1547	B Tools and Agents in FedAgentBench Framework	33
1548	B.1 Collection of Tools accessed by the LLM Agents	33
1549	B.2 Role-specialized Agents	42
1550		
1551		
1552	C Tasks and Algorithms in FedAgentBench Framework	49
1553	C.1 Dataset Details	49
1554	C.2 Sample dataset description files:	66
1555	C.3 Detecting and Addressing Data Quality Issues for Data Pre-Processing Agent . . .	78
1556	C.4 Collection of Federated Learning algorithms	79
1557	C.5 LLMs as the agent core components	88
1558		
1559		
1560		
1561	D Results and Discussions	89
1562	D.1 Discussion on agentic performance in individual healthcare environment	89
1563	D.2 Discussion on time-efficiency	96
1564	D.3 Discussion on client selection, reasoning vs non-reasoning models and failure modes: .	97
1565		

1566	D.4 Federated Training Performance	99
1567		
1568	E Future Work	100
1569		
1570	F Detailed insights from the Benchmark	101
1571		
1572	G Privacy Analysis of Harmonized Labels and Metadata	103
1573		
1574	G.1 Mutual Information Analysis	103
1575		
1576	G.2 Differential Privacy (DP) Proof	104
1577		
1578	G.3 k-Anonymity Analysis	104
1579		
1580	G.4 Privacy-Utility Trade-off	104
1581	H Broader Social Impact	114
1582		
1583	I LLM Usage:	114
1584		
1585		
1586		
1587		
1588		
1589		
1590		
1591		
1592		
1593		
1594		
1595		
1596		
1597		
1598		
1599		
1600		
1601		
1602		
1603		
1604		
1605		
1606		
1607		
1608		
1609		
1610		
1611		
1612		
1613		
1614		
1615		
1616		
1617		
1618		
1619		

1620 **A RELATED WORKS**
16211622 **A.1 FEDERATED LEARNING FOR MEDICAL IMAGE ANALYSIS**
16231624 Existing research on federated learning (FL) in medical image analysis has primarily focused on the
1625 development of machine learning algorithms to address technical challenges, such as data distribution
1626 shift, statistical and system heterogeneity, and communication efficiency (Antunes et al., 2022;
1627 Rajendran et al., 2021; Nguyen et al., 2022b; Pfitzner et al., 2021; Rieke et al., 2020). These efforts
1628 have produced a wide range of methods tailored for robust and scalable training under diverse and
1629 decentralized medical data environments. However, despite these advances, a significant barrier to
1630 real-world deployment persists: the complex set of operational and human-in-the-loop challenges
1631 encountered in practice.1632 Notably, existing FL benchmarks and studies rarely account for the intricacies of human factors—such
1633 as institutional workflows, task specification, annotation and curation requirements, and the expertise
1634 needed to orchestrate the entire FL pipeline across multiple healthcare institutions. These operational
1635 hurdles, including coordination among stakeholders, error handling, and workflow reproducibility,
1636 often constitute the most substantial obstacles to routine FL adoption in clinical settings.1637 This paper distinguishes itself from prior work by explicitly modeling and integrating these real-
1638 world operational challenges into the benchmarking process. By capturing both the algorithmic
1639 and human-centered aspects of FL deployment, our benchmark provides a more comprehensive and
1640 realistic evaluation platform. This enables the research community to move beyond algorithm-centric
1641 benchmarks and address the "elephant in the room", *i.e.*, the operational bottlenecks that ultimately
1642 determine the success or failure of federated learning in medical imaging practice.1643
1644 **A.2 LLM AGENT APPLICATIONS**
16451646 AI agents, powered by large language models (LLMs), autonomous tool use, and decision-making
1647 workflows, are rapidly transforming a diverse range of application domains. In **healthcare**, LLM-
1648 based agents drive advances in clinical diagnosis (Chen et al., 2024; Zhou et al., 2024; Wang et al.,
1649 2025c; Rose et al., 2025; Ghezloo et al., 2025; Li et al., 2024a; Jiang et al., 2025; Kim et al., 2024;
1650 Fallahpour et al.), mental health and therapy (Wasenmüller et al., 2024; Du et al., 2024; Zhang
1651 et al., 2024b; Lee et al., 2025; Xu et al., 2025a; Yang et al., 2025b; Steenstra et al., 2025; Abbasi
1652 et al., 2025), workflow optimization (Feng et al., 2025; Yun et al., 2025; Chen et al., 2025d), and
1653 pharmaceutical research (Wang et al., 2024c; Averly et al., 2025; Inoue et al., 2024). These agents
1654 support professionals through transparent reasoning, multi-modal data integration, and interactive,
1655 explainable decision support, as well as automated data processing and clinical research acceleration.1656 In **biomedical and materials science**, agents enhance literature analysis and hypothesis generation
1657 (Liang et al., 2025; Li et al., 2024b; Schmidgall & Moor, 2025; Gottweis et al., 2025), automate
1658 gene set knowledge discovery (Wang et al., 2024d), and orchestrate complex scientific workflows,
1659 including astronomical observation (Wang et al., 2024a) and materials design (Zhang et al., 2024a;
1660 Kumbhar et al., 2025).1661 The field of **software engineering** benefits from LLM agents for code generation, repair, verification,
1662 and environment setup (Dong, 2025; Jain et al., 2025; Wang et al., 2025a; Chen et al., 2025b;
1663 Aggarwal et al., 2025; Chen et al., 2025c; Gholamzadeh Khoei et al., 2025; Hu et al., 2025; Lu et al.,
1664 2025; Pan et al., 2024; Yang et al., 2025a; Guo et al., 2025; Islam et al., 2025). These agents leverage
1665 specialized architectures, collaborative multi-agent strategies, and benchmarking frameworks for
1666 automated programming, debugging, and user experience testing.1667 In **finance**, AI agents automate structured finance workflows, simulate markets, optimize investment
1668 decisions, and manage risk (Wan et al., 2024; Yang et al., 2025c; Yu et al., 2024b; Lin et al., 2024;
1669 Fatemi & Hu, 2024; Han et al., 2024b; 2025; Fatouros et al., 2025; Okpala et al., 2025; Zeng
1670 et al., 2025). Multi-agent frameworks enable complex reasoning, robust QA, and the generation of
1671 explainable financial reports.1672 **Synthetic data generation** is advanced through multi-agent orchestration frameworks (Mitra et al.,
1673 2024), improving post-training data quality and scalability for large language models.

1674 In **chemistry and materials**, agents automate chemical reasoning (Cho et al., 2025; Tang et al.,
 1675 2025), accelerate drug and materials discovery, and enable hypothesis-driven research (Zhang et al.,
 1676 2024a; Kumbhar et al., 2025).

1677 **Mathematics education and scientific reasoning** have seen the development of multi-agent rea-
 1678 soning and tutoring systems to tackle complex mathematical proofs, theorem proving, and adaptive
 1679 instruction (Lei et al., 2024; Xie et al., 2024; Lee et al., 2024; Deng & Mineiro, 2024; Li et al., 2025;
 1680 Wang et al., 2025b; Yue et al., 2024; Liu et al., 2025; Ma et al., 2025).

1681 In **geospatial science**, agents facilitate autonomous GIS analysis and data retrieval (Yu et al., 2024a;
 1682 Ning et al., 2025), addressing the challenge of spatial reasoning and multi-source data fusion.

1683 The domain of **multimedia and creative industries** is being transformed by AI agents capable of
 1684 automating film production, music and lyric generation, story-to-video creation, fashion assistance,
 1685 and poetry composition (Xu et al., 2025b; Wang et al., 2024b; Han et al., 2024a; Maronikolakis et al.,
 1686 2024; Deng et al., 2024; Yu et al., 2023; Zhang & Eger, 2024; Liu & Liu, 2024). These systems
 1687 support multi-modal content creation and human-AI co-creation.

1688 Overall, the emergence of LLM-powered agents marks a shift toward highly automated, context-
 1689 aware, and collaborative AI systems with applications spanning healthcare, science, engineering,
 1690 finance, education, and the creative arts.

1691 1692 1693 1694 1695 1696 A.3 LLM AGENTS FOR MACHINE LEARNING, SOFTWARE ENGINEERING, AND FEDERATED 1697 LEARNING

1698 1700 The intersection of large language models (LLMs) and autonomous agents has made rapid advance-
 1701 1702 ments in machine learning and software engineering. Several works (Chen et al., 2021; Hendrycks
 1703 et al., 2021; Austin et al., 2021; Jain et al., 2024) assess model performance on code generation from
 1704 natural language instructions. For example: AgentCoder (Huang et al., 2024a) reports 96.3% and
 1705 91.8% accuracy on HumanEval and MBPP, respectively. SWE-bench (Jimenez et al., 2024) advances
 1706 the field by requiring models to resolve real-world pull requests from open-source repositories.
 1707 Notably, model performance on SWE-bench continues to improve steadily (Zhang et al., 2024c;
 1708 factory.ai, 2024).

1709 1710 Prior work has also leveraged LLMs for tasks such as hyperparameter optimization (Liu et al., 2024b)
 1711 and neural architecture design (Zheng et al., 2023). MLAgentBench (Huang et al., 2024b) evaluates
 1712 agents on 13 Kaggle and custom ML tasks, providing a baseline solution for each and measuring
 1713 whether agents can achieve at least a 10% improvement. Similarly, ML-Bench (Tang et al., 2024)
 1714 evaluates an agent’s ability to generate code and interact with established ML repositories. AIDE,
 1715 as reported by Weco AI (Schmidt et al., 2024), surpasses more than 50% of human competitors in
 1716 Kaggle-style data science contests. DSbench (Jing et al., 2024) also introduces a Kaggle competition
 1717 benchmark, but, like Weco AI, focuses primarily on data science tasks.

1718 1719 While benchmarking LLM agents for automated machine learning and data science has gained
 1720 momentum across both academia and industry, all of these operate under the assumption of a
 1721 centralized, single-site environment, limiting their applicability to the federated learning paradigm,
 1722 which introduces unique challenges such as distributed data silos, partial observability, and multi-
 1723 party coordination. Recent works on agentic FL frameworks include in-context learning in FL of
 1724 LLM agents (Wu et al., 2024), reinforcement learning agent for client selection (Nasr & Hachaïchi),
 1725 and privacy enhancing techniques in federated multi-agent systems (Shi et al.).

1726 1727 In contrast to these works, **FedAgentBench** is designed to address the real-world operational com-
 1728 1729 plexities in federated learning workflows by evaluating the agentic capabilities — particularly in
 1730 high-stakes healthcare settings. **Rather than being “yet another” benchmark, FedAgentBench is
 1731 1732 motivated by a concrete and pressing need to reduce the human coordination bottlenecks that
 1733 currently hinder scalable deployment of FL in practice.** It provides a realistic testbed for assessing
 1734 agent autonomy, adaptability, and reasoning in decentralized, privacy-preserving environments.

1728 B TOOLS AND AGENTS IN FEDAGENTBENCH FRAMEWORK

1730 B.1 COLLECTION OF TOOLS ACCESSED BY THE LLM AGENTS

1732 The following tools form the operational backbone of the LLM-based agents, enabling tasks such
 1733 as file inspection, dataset organization, data cleaning, folder manipulation, and federated training
 1734 orchestration. Corresponding code snippets for all 16 tools can be found in Listing 1.

- 1736 1. **read_files**: Reads the content of one or more specified files and returns a dictionary mapping
 1737 file paths to their contents. It supports UTF-8 text files and handles file access errors
 1738 gracefully.
- 1739 2. **move_directory**: Moves a source directory (including all files and subfolders) to a new
 1740 destination.
- 1741 3. **copy_files**: Copies multiple individual files to specified destination paths. Accepts a mapping
 1742 of source to destination file paths and ensures target directories are created as needed.
- 1743 4. **write_file**: Writes a given text string to a specified file path. It creates any missing directories
 1744 in the path before writing.
- 1745 5. **edit_file**: Overwrites the contents of a specified file with new content. Used for completely
 1746 replacing existing file content.
- 1747 6. **run_script**: Executes a given shell command (typically a Python script) using a secure
 1748 subprocess or shell tool backend. Returns the result of the command execution.
- 1749 7. **list_files_in_second_level**: Traverses the second-level entries of a root directory. For each
 1750 subdirectory or file, it collects and returns metadata including the total number of files and a
 1751 preview list of file paths (up to 10).
- 1752 8. **preview_file_content**: Previews the contents of a CSV, JSON, or TXT file. Returns first 5
 1753 rows or entries and summary statistics such as total rows or elements.
- 1754 9. **run_selfclean_on_dataset**: Runs the data cleaning framework on an image folder to detect
 1755 and optionally clean near duplicates, off-topic or irrelevant samples, and label errors. It
 1756 generates internal diagnostic data in CSV format for inspection and removes samples based
 1757 on a threshold. Within this process, we also achieve normalization and standardization.
- 1758 10. **organize_into_subfolder**: Reads a CSV containing image paths and labels, and organizes
 1759 the corresponding images into class-specific subfolders within a specified destination
 1760 directory.
- 1761 11. **copy_folder**: Copies all contents (files and subfolders) from a source directory to a destination
 1762 directory. Ensures destination exists and performs a recursive copy.
- 1763 12. **remove_other_files**: Recursively removes all non-image files from a directory structure.
 1764 Keeps standard image formats (e.g., .jpg, .png, .bmp) and deletes all others.
- 1765 13. **list_folders**: Returns the names of all first-level subdirectories under a specified root
 1766 directory. Useful for summarizing dataset structure.
- 1767 14. **make_folder**: Creates a new directory at a specified path. Used to set up target folders
 1768 during label harmonization or preprocessing.
- 1769 15. **copy_images**: Copies all image files from a source folder to a specified target folder.
 1770 Typically used during label harmonization to reorganize class-wise images.
- 1771 16. **run_federated_method**: Launches federated learning using a specified algorithm and
 1772 project directory. Executes a Python script with algorithm-specific parameters and returns
 1773 algorithm performance.

1776 Listing 1: Repository of tools used by LLM Agents

```
1777 1. def read_files(file_paths: list) -> dict:
1778     """
1779     Read file contents and return as dictionary.
1780
1781     Args:
1782         file_paths: List of file paths to read
```

```

1782
1783     Returns:
1784         dict: Dictionary with {file_path: file_content} format
1785         """
1786     file_contents = {}
1787
1788     for file_path in file_paths:
1789         try:
1790             with open(file_path, 'r', encoding='utf-8') as file:
1791                 content = file.read()
1792                 file_contents[file_path] = content
1793         except (UnicodeDecodeError, PermissionError, FileNotFoundError):
1794             as e:
1795                 print(f"Cannot read file {file_path}: {e}")
1796                 file_contents[file_path] = None
1797
1798     return file_contents
1799
1800 2. def move_directory(src_dir: str, dest_dir: str) -> str:
1801     """
1802         Move source directory and its contents to destination directory,
1803         creating a new subdirectory
1804         with the same name as the source directory.
1805
1806     Args:
1807         src_dir: Source directory path (e.g., '/path/to/source/
1808                 folder_name')
1809         dest_dir: Parent destination directory path (e.g., '/path/to/dest
1810                 ')
1811                 A new subdirectory named 'folder_name' will be created
1812                 here
1813
1814     Returns:
1815         str: Operation result message
1816
1817     Example:
1818         If src_dir is '/path/to/source/folder_name' and dest_dir is '/
1819             path/to/dest',
1820             the directory will be moved to '/path/to/dest/folder_name'
1821         """
1822     print(f"Running move_directory tool to move from {src_dir} to {
1823         dest_dir}...")
1824     try:
1825         if not os.path.exists(src_dir):
1826             return f"Source directory {src_dir} does not exist"
1827
1828         # Get the source directory name
1829         src_name = os.path.basename(src_dir.rstrip('/'))
1830         target_dir = os.path.join(dest_dir, src_name)
1831
1832         # If destination directory already exists, remove it first
1833         if os.path.exists(target_dir):
1834             shutil.rmtree(target_dir)
1835
1836         # Move the directory
1837         shutil.move(src_dir, target_dir)
1838         return f"Directory {src_dir} has been successfully moved to {
1839             target_dir}"
1840
1841     except Exception as e:
1842         return f"Error moving directory: {str(e)}"
1843
1844
1845 3. def copy_files(file_mapping: dict) -> str:
1846     """

```

```

1836     Copy multiple files from source paths to destination paths.
1837
1838     Args:
1839         file_mapping (dict): A dictionary where keys are source file
1840                         paths and values are destination file paths.
1841         Example:
1842             "/path/to/source1.txt": "/path/to/destination1.txt",
1843             "/path/to/source2.txt": "/path/to/destination2.txt"
1844
1845     Returns:
1846         str: A message indicating the result of the operation.
1847         """
1848     print(f"Running copy_files tool to copy {file_mapping}...")
1849     results = []
1850     for src, dest in file_mapping.items():
1851         try:
1852             # Check if source file exists
1853             if not os.path.exists(src):
1854                 results.append(f"Source file {src} does not exist.")
1855                 continue
1856
1857             # Create destination directory if it doesn't exist
1858             dest_directory = os.path.dirname(dest)
1859             if not os.path.exists(dest_directory):
1860                 os.makedirs(dest_directory)
1861
1862             # Copy file
1863             shutil.copy2(src, dest)
1864             results.append(f"File {src} successfully copied to {dest}")
1865
1866     # Return summary of all operations
1867     return "\n".join(results)
1868
1869 4. def write_file(content: str, file_path: str) -> None:
1870     """
1871     Write a given string of code to a specified file.
1872
1873     This function creates the necessary directories for the file (if they
1874     don't exist),
1875     writes the content to the file, and handles any errors that may occur
1876     during the process.
1877
1878     Args:
1879         content (str): The code or text you want to write into the file.
1880         file_path (str): The full path (including filename) where the
1881                         content will be saved.
1882
1883     Example:
1884         write_file('print("Hello World")', 'scripts/hello.py')
1885         """
1886     print(f"Running write_file tool to write {file_path}...")
1887     try:
1888         os.makedirs(os.path.dirname(file_path), exist_ok=True)
1889
1890         with open(file_path, 'w', encoding='utf-8') as file:
1891             file.write(content)
1892
1893             print(f"File successfully written to: {file_path}")
1894     except Exception as e:
1895         print(f"Error writing file: {e}")

```

```

1890
1891 5. def edit_file(new_content: str, file_path: str) -> None:
1892     """
1893         Completely overwrite a file with new content. The original file
1894         content will be replaced entirely.
1895
1896     Args:
1897         new_content: Complete content to replace the existing file
1898             content. This should be the entire
1899                 desired content of the file after editing, not just
1900                     the changes.
1901         file_path: Path of the file to edit
1902
1903     Note:
1904         This function performs a complete overwrite operation. The
1905             original content will be lost.
1906         You must provide the complete desired final content, including
1907             both modified and unmodified parts.
1908         """
1909     print(f"Running edit_file tool to edit {file_path}...")
1910     try:
1911         with open(file_path, 'w', encoding='utf-8') as file:
1912             file.write(new_content)
1913
1914     print(f"File {file_path} successfully edited.")
1915     except Exception as e:
1916         print(f"Error editing file: {e}")
1917
1918 6. def run_script(command: str) -> str:
1919     """
1920         Execute shell command
1921
1922     Args:
1923         command: Shell command to execute
1924
1925     Returns:
1926         str: Command execution result
1927     """
1928     cmd_base, script_path = command.strip().split(maxsplit=1)
1929
1930     # Blindly quote the path
1931     script_path = f'"{script_path}"'
1932
1933     # Rebuild the final command
1934     fixed_command = f'{cmd_base} {script_path}'
1935
1936     print(f"Executing fixed command: {fixed_command}")
1937     print("Running run_script tool...")
1938     shell_tool = ShellTool()
1939     result = shell_tool.run({
1940         "commands": [fixed_command]
1941     })
1942     return result
1943
1944 def natural_sort_key(s):
1945     """
1946         Generate a key for natural sorting.
1947
1948         This function splits the string into numeric and non-numeric parts so
1949             that,
1950             for example, "file2" is sorted before "file10".
1951     """
1952     return [int(text) if text.isdigit() else text.lower() for text in re.
1953             split(r'(\d+)', s)]

```

```

1944 def get_second_level_entries(root_dir):
1945     """
1946     Retrieve all second-level entries (files and directories) under the
1947     specified root directory,
1948     and sort them so that directories come first, then files. Both are
1949     sorted naturally.
1950     """
1951     try:
1952         entries = list(os.scandir(root_dir))
1953     except Exception as e:
1954         print(f"Error scanning {root_dir}: {e}")
1955         return []
1956
1957     entries.sort(key=lambda e: (not e.is_dir(), natural_sort_key(e.name)))
1958
1959     return entries
1960
1961 def collect_all_files_from_directory(directory):
1962     """
1963     Recursively collect all file paths from the given directory,
1964     sorted naturally by their relative paths.
1965     """
1966     collected = []
1967     for root, dirs, files in os.walk(directory):
1968         dirs.sort(key=natural_sort_key)
1969         files.sort(key=natural_sort_key)
1970         for file in files:
1971             full_file_path = os.path.join(root, file)
1972             relative_path = os.path.relpath(full_file_path, start=
1973                 directory)
1974             collected.append((relative_path, full_file_path))
1975     collected.sort(key=lambda tup: natural_sort_key(tup[0]))
1976     return collected
1977
1978 def list_files_in_second_level(root_directory: str) -> dict:
1979     """
1980     Traverse all second-level entries under the root directory and return
1981     a summary dictionary.
1982     """
1983     print(f"Running list_files_in_second_level tool under {root_directory}
1984         ...")
1985     max_files = 10
1986     results = []
1987     second_level_entries = get_second_level_entries(root_directory)
1988
1989     for entry in second_level_entries:
1990         if entry.is_file():
1991             result_dict = {
1992                 "entry_name": entry.name,
1993                 "entry_path": entry.path,
1994                 "total_files": 1,
1995                 "files": [entry.path]
1996             }
1997             results.append(result_dict)
1998         elif entry.is_dir():
1999             collected_files = collect_all_files_from_directory(entry.path)
2000             total_file_count = len(collected_files)
2001             top_files = [full_path for _, full_path in collected_files[:-
2002                 max_files]]
2003             result_dict = {
2004                 "entry_name": entry.name,
2005                 "entry_path": entry.path,
2006                 "total_files": total_file_count,
2007                 "files": top_files
2008             }
2009             results.append(result_dict)
2010
2011     return results

```

```

1998         }
1999     results.append(result_dict)
2000
2001     final_result = {"entries": results}
2002     print(final_result)
2003     return final_result
2004
2005     8. def preview_file_content(file_path: str) -> str:
2006         """
2007             Preview the contents of CSV, JSON, or TXT files.
2008         """
2009         print(f"Running preview_file_content tool for {file_path}...")
2010         if file_path.lower().endswith('.csv'):
2011             rows = []
2012             total_rows = 0
2013             try:
2014                 with open(file_path, 'r', encoding='utf-8') as f:
2015                     reader = csv.reader(f)
2016                     for row in reader:
2017                         total_rows += 1
2018                         if total_rows <= 5:
2019                             rows.append(row)
2020             except Exception as e:
2021                 return f"Error reading CSV file: {e}"
2022
2023             preview_str = "CSV File Preview:\n"
2024             for row in rows:
2025                 preview_str += ", ".join(row) + "\n"
2026             preview_str += f"Total rows: {total_rows}"
2027             return preview_str
2028
2029         elif file_path.lower().endswith('.json'):
2030             try:
2031                 with open(file_path, 'r', encoding='utf-8') as f:
2032                     data = json.load(f)
2033             except Exception as e:
2034                 return f"Error reading JSON file: {e}"
2035
2036             if isinstance(data, dict):
2037                 items = list(data.items())
2038                 preview_items = items[:5]
2039                 preview_str = "JSON File Preview (first 5 key-value pairs):\n"
2040
2041                 for key, value in preview_items:
2042                     preview_str += f"{key}: {value}\n"
2043                 preview_str += f"Total key-value pairs: {len(items)}\n"
2044             elif isinstance(data, list):
2045                 preview_items = data[:5]
2046                 preview_str = "JSON File Preview (first 5 elements):\n"
2047                 for item in preview_items:
2048                     preview_str += f"{item}\n"
2049                 preview_str += f"Total elements: {len(data)}\n"
2050             else:
2051                 preview_str = f"Unsupported JSON type: {type(data)}\n"
2052             return preview_str
2053
2054         elif file_path.lower().endswith('.txt'):
2055             try:
2056                 with open(file_path, 'r', encoding='utf-8') as f:
2057                     content = f.read()
2058             except Exception as e:
2059                 return f"Error reading TXT file: {e}"
2060
2061             words = content.split()
2062             total_words = len(words)

```

```

2052     preview_words = words[:10000]
2053     preview_str = "TXT File Preview (first 10000 words):\n"
2054     preview_str += " ".join(preview_words)
2055     preview_str += f"\nTotal words: {total_words}"
2056     return "==== CSV Preview ===\n" + preview_str
2057
2058     else:
2059         return "Unsupported file type. Only CSV, JSON, and TXT files are
2060         supported."
2061
2062 9. def run_selfclean_on_dataset(image_folder_path: str) -> None:
2063     """
2064     Run SelfClean on an image folder and generate CSVs for near
2065     duplicates, off-topic samples, and label errors.
2066
2067     Args:
2068         image_folder_path (str): Path to the root folder containing the
2069             images organized by class folders.
2070     """
2071     sc_utils.init_distributed_mode = dummy_init_distributed_mode
2072
2073     # Patch torch.load for compatibility
2074     original_torch_load = torch.load
2075     def patched_torch_load(*args, **kwargs):
2076         kwargs["weights_only"] = False
2077         return original_torch_load(*args, **kwargs)
2078     torch.load = patched_torch_load
2079
2080     resize_images_in_folder(image_folder_path)
2081
2082     print("Loading dataset with ImageFolder...")
2083     dataset = ImageFolder(root=image_folder_path)
2084
2085     parameters = copy.deepcopy(DINO_STANDARD_HYPERPARAMETERS)
2086     parameters['model']['base_model'] = 'pretrained_imagenet_vit_tiny'
2087
2088     print("Running SelfClean...")
2089     selfclean = SelfClean(auto_cleaning=True)
2090     print("Selfclean loaded")
2091
2092     def patched_load_pretrained(model_name=None, work_dir=None, **kwargs):
2093         print("Using locally downloaded DINO checkpoint")
2094         local_model_path = "path/to/model"
2095         model = sc_utils.Embedder.load_dino(ckp_path=local_model_path)
2096         dummy_config = SimpleNamespace(model_type="ViT")
2097         dummy_augment_fn = lambda x: x
2098         return model, dummy_config, dummy_augment_fn
2099     sc_utils.Embedder.load_pretrained = patched_load_pretrained
2100
2101     work_folder_path = {"..."} .get(image_folder_path, None)
2102
2103     issues = selfclean.run_on_dataset(
2104         dataset=copy.copy(dataset),
2105         pretraining_type=PretrainingType.DINO,
2106         epochs=10,
2107         batch_size=16,
2108         save_every_n_epochs=1,
2109         dataset_name="...",
2110         work_dir=work_folder_path,
2111     )
2112
2113     df_near_duplicates = issues.get_issues("near_duplicates",
2114                                         return_as_df=True)

```

```

2106
2107     df_off_topic_samples = issues.get_issues("off_topic_samples",
2108         return_as_df=True)
2109     df_label_errors = issues.get_issues("label_errors", return_as_df=True
2110         )
2111
2112 10. def organize_into_subfolder(root_directory: str,
2113     destination_directory: str) -> dict:
2114     """
2115     Organize images into class-wise subfolders using labels from a CSV
2116     file.
2117     """
2118     try:
2119         csv_files = [f for f in os.listdir(root_directory) if f.endswith(
2120             ".csv")]
2121         if len(csv_files) != 1:
2122             return {"status": "error", "message": "Expected exactly one
2123             CSV file."}
2124
2125         csv_path = os.path.join(root_directory, csv_files[0])
2126         df = pd.read_csv(csv_path)
2127
2128         label_col = [col for col in df.columns if "label" in col.lower()
2129             ][0]
2130         file_col = [col for col in df.columns if "file" in col.lower() or
2131             "image" in col.lower() or "path" in col.lower()][0]
2132
2133         moved_count = {}
2134         for _, row in df.iterrows():
2135             label = str(row[label_col]).strip()
2136             filename = str(row[file_col]).strip()
2137             src_path = filename
2138             if not os.path.exists(src_path):
2139                 continue
2140
2141             label_folder = os.path.join(destination_directory, label)
2142             os.makedirs(label_folder, exist_ok=True)
2143             dst_path = os.path.join(label_folder, os.path.basename(
2144                 filename))
2145             shutil.copy2(src_path, dst_path)
2146             moved_count[label] = moved_count.get(label, 0) + 1
2147
2148             return {"status": "success", "moved": moved_count}
2149         except Exception as e:
2150             return {"status": "error", "message": str(e)}
2151
2152 11. def copy_folder(source_directory: str, destination_directory: str) ->
2153     dict:
2154     """
2155     Copies all files and subdirectories from source to destination.
2156     """
2157     try:
2158         if not os.path.exists(source_directory):
2159             return {"status": "error", "message": f"Source folder does
2160                 not exist: {source_directory}"}
2161             os.makedirs(destination_directory, exist_ok=True)
2162
2163             for item in os.listdir(source_directory):
2164                 src = os.path.join(source_directory, item)
2165                 dst = os.path.join(destination_directory, item)
2166                 if os.path.isdir(src):
2167                     shutil.copytree(src, dst, dirs_exist_ok=True)
2168                 else:
2169                     shutil.copy2(src, dst)

```

```

2160
2161     return {"status": "success", "message": f"Copied from {source_directory} to {destination_directory}"}
2162 except Exception as e:
2163     return {"status": "error", "message": str(e)}
2164
2165
2166 12. def remove_other_files(root_directory: str) -> dict:
2167     """
2168     Remove all non-image files from a directory and its subdirectories.
2169     """
2170     allowed_extensions = {'.jpg', '.jpeg', '.png', '.bmp', '.tiff', '.tif',
2171                           ', '.gif', '.dcm', '.nii', '.nii.gz', '.mha', '.mhd', '.hdr', '.img',
2172                           '.nrrd'}
2173     removed_files = []
2174
2175     for dirpath, _, filenames in os.walk(root_directory):
2176         for filename in filenames:
2177             ext = os.path.splitext(filename)[1].lower()
2178             if ext not in allowed_extensions:
2179                 file_path = os.path.join(dirpath, filename)
2180                 try:
2181                     os.remove(file_path)
2182                     removed_files.append(file_path)
2183                 except Exception as e:
2184                     print(f"Error removing {file_path}: {e}")
2185
2186     return {"status": "success", "removed_file_count": len(removed_files),
2187             "removed_files": removed_files}
2188
2189
2190 13. def list_folders(root_directory: str) -> dict:
2191     """
2192     List subfolders in the given directory.
2193     """
2194     folders = [f for f in os.listdir(root_directory) if os.path.isdir(os.path.join(root_directory, f))]
2195     return {"folders": folders}
2196
2197
2198 14. def make_folder(root_directory: str) -> dict:
2199     """
2200     Create a new folder at the given path.
2201     """
2202     try:
2203         os.makedirs(root_directory, exist_ok=True)
2204         return {"status": "success", "message": f"Created folder: {root_directory}"}
2205     except Exception as e:
2206         return {"status": "error", "message": str(e)}
2207
2208
2209 15. def copy_images(src_folder: str, dst_folder: str) -> dict:
2210     """
2211     Copies all image files from the source folder (including subfolders)
2212     to the destination folder.
2213
2214     Args:
2215         src_folder (str): Path to the source folder containing image
2216                         files.
2217         dst_folder (str): Path to the destination folder where images
2218                         will be copied.
2219
2220     Returns:
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
433
```

B.2 ROLE-SPECIALIZED AGENTS

To enable automated, modular, and scalable orchestration of federated learning workflows, we introduce a suite of seven specialized LLM agents within the FedAgentBench framework. Each

2268 agent is assigned a distinct responsibility aligned with a specific stage of the FL pipeline, spanning
 2269 from task interpretation and dataset selection to data preparation, label harmonization, algorithm
 2270 selection, and training. These agents collectively simulate the collaborative behavior typically
 2271 required from domain experts, data engineers, and FL researchers, while interacting through well-
 2272 defined prompts and toolchains. Code snippets of all 7 role-specialized agents can be found in
 2273 Listings 2-5 with each discussing agents of individual phases.

2274

2275 **RESPONSIBILITIES OF FEDAGENTBENCH AGENTS:**

2276

2277 As a part of FedAgentBench, we design a modular and collaborative framework composed of seven
 2278 specialized LLM agents, each responsible for a distinct role in the federated learning pipeline and
 2279 operating via specific toolsets (if necessary) that allow them to automate key stages of client-server
 2280 coordination, data preparation, and model training. Table 3 summarizes the roles of the seven
 2281 specialized agents. Below, we describe the function of each agent in the context of the four major
 2282 phases of the workflow.

2283

1. **Server Agent for Task Interpretation (S_1):** This agent parses the user-defined instruction to identify the intended task and required data modality. It then broadcasts this extracted requirement to all client agents to begin the dataset discovery process.
2. **Client Selector Agent (C_1):** After receiving the task description from the server, this agent inspects the metadata of available datasets and determines which of them are relevant to the given task. The selection is based on textual descriptions stored in a structured JSON file. This task is facilitated using the `read_files` function to analyze the dataset content. The agent responds with matching dataset names or returns "no dataset" if none are suitable.
3. **Server Agent for Client Approval (S_2):** This agent is responsible for validating the responses returned by the client agents. If a client proposes one or more datasets, the server responds with "Approved. Prepare for training". If the client has no relevant data, the server sends "Client not needed for the task" to exclude them from training.
4. **Data Pre-processor Agent (C_2):** This agent ensures the dataset is well-organized and free from noisy or irrelevant samples. It first checks whether the dataset is structured in class-specific subfolders. If not, it reorganizes the data accordingly. It then eliminates all non-image files and performs content-based cleaning to flag duplicates, off-topic, or mislabeled samples. These operations can be carried out using tools such as `organize_into_subfolder`, `remove_other_files`, and `run_selfclean_on_dataset` discussed earlier. The agent concludes by signaling completion with "Data Cleaning Complete <end>".
5. **Task conditioned Label Harmonizer Agent (C_3):** This agent unifies the class label space across multiple clients by remapping existing class folders into a shared label schema (e.g., from fine-grained categories to binary classes like malignant or benign). It first lists the current folder names, defines a harmonization mapping, and creates new folders to reflect the harmonized schema. This can be accomplished using `list_folders`, `make_folder`, and `copy_images` functions mentioned earlier.
6. **FL Algorithm Selector Agent (S_3):** This agent chooses the most appropriate federated learning algorithm for training based on the user's task requirement. It examines a JSON file describing available algorithms and selects one based on the alignment of its key idea and name with the user's intent. This process can be supported by the `read_files` tool and results in a response such as "Algorithm Name: ... <end>".
7. **Trainer Agent (S_4):** Once the data and algorithm are finalized, this agent launches federated training using the selected method. It delegates execution to the appropriate script that implements the algorithm. This can be done by calling the `run_federated_method` tool.

2319

2320 **Justification of Agent Design.** The decomposition into seven specialized agents is grounded in
 2321 the need to modularize a complex and multi-phase federated learning pipeline that must accommodate the broad diversity of FL algorithms (as evidenced in FL-Bench, spanning aggregation-based,

personalization-based, and representation-based strategies) and ensure automation across heterogeneous datasets and institutional constraints. The separation of concerns allows each agent to handle a distinct phase of the workflow: high-level task parsing (S_1), distributed dataset discovery (C_1), client validation (S_2), data reorganization and quality control (C_2), cross-client label harmonization (C_3), FL algorithm selection conditioned on user intent (S_3), and training orchestration (S_4). This division aligns with the key bottlenecks in real-world FL deployment. The agent specialization ensures scalability, adaptability, and plug-and-play extensibility of the framework, enabling future integration of additional FL capabilities (e.g., fairness, security, cross-silo adaptation) without architectural redesign. The code snippets of the individual specialized agents are provided below:

CODE SNIPPETS OF SPECIALIZED AGENTS:

Listing 2: Prompt definition for Client Orchestrator Agents

```

2335 def create_server_to_client_communication_prompt_round_1():
2336     system_prompt = """
2337     You are a server agent in a Federated Learning setup, responsible for
2338         communicating with the client agents.
2339     From the user requirement, only extract the task and modality
2340         information.
2341     State this information and instruct the clients to respond with:
2342         - The name of the selected dataset (that matches the user requirement
2343             )
2344     """
2345     return system_prompt
2346
2347 # Goal-oriented guidance
2348 def create_selector_prompt(description_path, server_instruction):
2349     system_prompt = f"""
2350     You are acting as a client agent in Federated Learning responsible
2351         for selecting the datasets in your client based on the server
2352             instructions: {server_instruction}.
2353     I provide you with a list of dataset descriptions: {description_path}
2354         , which is a json file that contains a list of dictionaries.
2355     Plan your workflow and solve the task:
2356
2357     You have access to the tool:
2358     read_files: This function reads a script file (such as a Python file)
2359         so you can understand its content.
2360
2361     Return the chosen dataset names following {server_instruction}, so a
2362         downstream peer agent can know the information accurately.
2363     IMPORTANT: Give it only in this template for each dataset: **Dataset
2364         Name** : .... If no suitable dataset for the given task exists,
2365             the client should return "no dataset" and clearly explain why
2366             before ending the conversation.
2367     Include <end> to end the conversation.
2368     """
2369     return system_prompt
2370
2371 # Fine-grained guidance
2372 def create_selector_prompt(description_path, server_instruction):
2373     system_prompt = f"""
2374     You are acting as a client agent in Federated Learning responsible
2375         for selecting the datasets in your client based on the server
2376             instructions: {server_instruction}.
2377     I provide you with a list of dataset descriptions: {description_path}
2378         , which is a json file that contains a list of dictionaries.
2379     Every dictionary contains following entries: ["Dataset
2380         Name", "Dataset Description", "dataset_path"].
2381
2382     You have access to the tools:
2383     read_files: This function reads a script file (such as a Python file)
2384         so you can understand its content.

```

```

2376
2377     Here is the typical workflow you should follow:
2378     1. Use read_files to read {description_path}, understand its content.
2379     2. Choose all the datasets that match the server instructions.
2380         Remember, your choice should be mainly based on "dataset
2381         descriptions" entry.
2382     3. Return the chosen dataset names following {server_instruction}, so
2383         a downstream peer agent can know the information accurately.
2384         IMPORTANT: Give it only in this template for each dataset: **Dataset
2385             Name** : .... If no suitable dataset for the given task exists,
2386             the client should return "no dataset" and clearly explain why
2387             before ending the conversation.
2388     4. Include <end> to end the conversation.
2389     """
2390     return system_prompt
2391
2392     def create_server_to_client_communication_prompt_round_2(client_response):
2393         :
2394         system_prompt = f"""
2395             You are acting as a server agent for communicating with the client
2396                 agents in Federated Learning. Read the client response: {
2397                     client_response}
2398             If the client has returned one or more datasets, return the message:
2399                 "Approved. Prepare for training".
2400             If the client has returned no dataset, return the message: "Client
2401                 not needed for the task".
2402             """
2403         return system_prompt
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429

```

Listing 3: Prompt definition for Data Pre-processor Agent

```

2402     # Goal-oriented guidance
2403     def create_datacleaner_prompt(input_data_path, output_data_path,
2404         server_response_round_2, description_path):
2405         system_prompt = f"""
2406             You are a highly skilled data preparation and data cleaning agent
2407                 specializing in the medical domain. You MUST do your tasks ONLY
2408                 using the tools provided to you.
2409             You MUST plan the workflow based on the instruction given below
2410                 sincerely and not bypass it.
2411             I provide you with server instruction {server_response_round_2}.
2412             If the server mentions that the client is not needed, end the
2413                 conversation and do NOT do anything else. Instead, if it
2414                 instructs to prepare for training, you have three tasks:
2415                 1. Check if the dataset in {input_data_path} is already organized in
2416                     sub-folder format from dataset descriptions: {description_path}.
2417                     If not, organize the data by grouping images of each class into
2418                         their respective subfolders in your destination path: {
2419                             output_data_path}.
2420                 2. Remove all non-image files from each sub-folder.
2421                 3. Clean client data by removing (a) near duplicate samples, (b) off
2422                     topic samples, (c) noisy label samples
2423
2424             You have access to the following tools. Plan and reason how to use
2425                 the following tools properly:
2426                 read_files: This function reads a script file (such as a Python file)
2427                     so you can understand its content.
2428                 organize_into_subfolder: This function reads csv file, goes through
2429                     the labels column, creates subfolders and groups images inside
2430                     them based on labels column.
2431                 copy_folder: This function copies folder from source location to
2432                     destination location.
2433                 remove_other_files: This function checks the file extension of all
2434                     files in a given folder and deletes the files with non-image
2435                     extensions.
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
2999

```

```

2430
2431     run_selfclean_on_dataset: This function flags (a) near duplicate
2432         samples, (b) off topic samples, (c) noisy label samples. Use this
2433         to clean the dataset
2434
2435     Important rules you must follow:
2436     - You MUST use the run_selfclean_on_dataset tool to clean data!
2437     - You MUST NOT modify the raw images manually.
2438     - You MUST conclude your work by writing: "Data Cleaning Complete" <
2439         end>.
2440     """
2441
2442     return system_prompt
2443
2444 # Fine-grained guidance
2445 def create_datacleaner_prompt(input_data_path, output_data_path,
2446     server_response_round_2, description_path):
2447     system_prompt = f"""
2448     You are a highly skilled data preparation and data cleaning agent
2449         specializing in the medical domain. I provide you with server
2450         instruction {server_response_round_2}.
2451     If the server mentions that the client is not needed, end the
2452         conversation. If it instructs to prepare for training, you have
2453         three tasks:
2454     1. Check if the dataset in {input_data_path} is already organized in
2455         sub-folder format from dataset descriptions: {description_path}.
2456         If not, organize the data by grouping images of each class into
2457             their respective subfolders in your destination path: {  

2458                 output_data_path}.
2459     2. Remove all non-image files from each sub-folder.
2460     3. Clean client data by removing (a) near duplicate samples, (b) off
2461         topic samples, (c) noisy label samples
2462
2463     You have access to the tools:
2464     read_files: This function reads a script file (such as a Python file)
2465         so you can understand its content.
2466     organize_into_subfolder: This function reads csv file, goes through
2467         the labels column, creates subfolders and groups images inside
2468         them based on labels column.
2469     copy_folder: This function copies folder from source location to
2470         destination location.
2471     remove_other_files: This function checks the file extension of all
2472         files in a given folder and deletes the files with non-image
2473         extensions.
2474     run_selfclean_on_dataset: This function flags (a) near duplicate
2475         samples, (b) off topic samples, (c) noisy label samples. Use this
2476         to clean the dataset
2477     clean_data: This function checks flagged samples from csv file and
2478         removes them.
2479
2480     Here is the typical workflow you should follow:
2481     1. If the server instruction: {server_response_round_2} mentions that
2482         the client is not needed, print <end> and end the conversation.
2483         Do NOT do anything further.
2484     2. Instead, if it instructs you to prepare for training, use "
2485         read_files" function to read and understand the dataset
2486         description file in {description_path}. Check from there, if the
2487         dataset in {input_data_path} is already organized as sub-folders.
2488         If yes, copy the folder to the destination folder {  

2489             output_data_path} using the function "copy_folder" and go to
2490             step 4 below, skipping step 3.
2491     3. If dataset is not organized as sub-folders, organize the data by
2492         grouping images of each class into their respective subfolders in
2493         the destination data path: {output_data_path} by using the
2494         organize_into_subfolder function.
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
3999

```

```

2484     4. Go to each subfolder in the destination data path: {
2485         output_data_path} and remove all non-image files by using
2486         remove_other_files function.
2487     5. Flag (a) near duplicate samples, (b) off topic samples, (c) noisy
2488         label samples using run_selfclean_on_dataset function.
2489     6. Remove the flagged samples using clean_data function.

2490     Important rules you must follow:
2491     - You MUST use the run_selfclean_on_dataset tool to clean data!
2492     - You MUST NOT modify the raw images manually.
2493     - You MUST clean using the CSV outputs only.
2494     - You MUST conclude your work by writing: "Data Cleaning Complete" <
2495         end>.
2496     """
2497     return system_prompt
2498
2499

```

Listing 4: Prompt definition for Label Harmonization Agent

```

2500 # Goal-oriented guidance
2501 def label_harmonizer_prompt(input_data_path, output_data_path):
2502     system_prompt = f"""
2503     You are an intelligent agent tasked with harmonizing medical image
2504         labels in a Federated Learning environment.
2505
2506     Your objective is to reorganize the dataset located at {
2507         input_data_path} by grouping existing class folders into
2508         standardized, harmonized categories (e.g., 'malignant', 'benign')
2509         based on the task specification.
2510
2511     You should inspect the current folder structure, define appropriate
2512         label mappings to target categories, and reorganize the data into
2513         the {output_data_path} directory using the available tools.
2514
2515     You have access to the following tools:
2516     - list_folders(path): Lists existing class folders in a dataset.
2517     - make_folder(path): Creates a new folder for a target label.
2518     - copy_images(src_folder, dst_folder): Copies all image files from
2519         the original to the harmonized destination folder.
2520
2521     Use these tools to achieve the goal of producing a clean, consistent
2522         label space for downstream federated training.
2523     When harmonization is complete, end your process with "<end>".
2524     """
2525     return system_prompt
2526
2527 # Fine-grained guidance
2528 def label_harmonizer_prompt(input_data_path, output_data_path):
2529     system_prompt = f"""
2530     You are an intelligent agent for medical image label harmonization in
2531         a Federated Learning setup.
2532     Your goal is to group existing class folders into harmonized target
2533         categories (e.g., 'malignant', 'benign') by reorganizing the
2534         folder structure.
2535     This involves identifying the current class folders, mapping them to
2536         new target labels, and copying images accordingly.
2537
2538     You have access to the tools:
2539     - list_folders(path): Returns a list of subfolder names in the given
2540         path.
2541     - make_folder(path): Creates a new directory at the specified path.
2542     - copy_images(src_folder, dst_folder): Copies all image files from
2543         the source to the destination folder.
2544
2545     Here is the typical workflow you should follow:

```

```

2538     1. Inspect class structure: Use 'list_folders("{input_data_path}")' ` 
2539        to get all existing class folder names.
2540     2. Define label mapping: Based on user requirements (e.g., binary
2541        classification), decide how existing class names map to target
2542        classes (coarse labels like 'malignant' and 'benign').
2543     3. Prepare new folders: For each target class, use 'make_folder("{"
2544        output_data_path}/<class_name>")' to create destination folders.
2545     4. Move data: For each source class, use 'copy_images' to move all
2546        image files to their new harmonized folder.
2547    """
2548
2549
2550
2551     return system_prompt
2548
2549
2550
2551

```

Listing 5: Prompt definition for Federated Trainer Agents

```

2552 # Goal-oriented guidance
2553 def FL_algorithm_selector_prompt(algorithm_description_path):
2554     system_prompt = f"""
2555     You are a server agent in a Federated Learning setup responsible for
2556        selecting the most appropriate federated learning algorithm based
2557        on the human user's task requirement.
2558
2559     You are provided with a list of algorithm descriptions in the file { 
2560        algorithm_description_path}, formatted as a JSON list of
2561        dictionaries. Each dictionary contains information about an
2562        algorithm, including its name, full name, and key idea.
2563
2564     Your objective is to analyze the algorithm descriptions and identify
2565        the method that best aligns with the user's intent. Focus
2566        primarily on the "Full Name" and "Key idea" fields to determine
2567        relevance.
2568
2569     You have access to the following tool:
2570     - read_files: This function reads a script file (such as a Python
2571        file) so you can understand its content.
2572
2573     Once you have selected the most suitable algorithm, return it in the
2574        format:
2575     Algorithm Name: <selected_algorithm>
2576
2577     Conclude your response with "<end>".
2578 """
2579
2580     return system_prompt
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3239
3240
3241
3242
3243
3244
3245
3246
3247
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3256
3257
3258
3259
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3398
3399
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3498
3499
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3598
3599
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3698
3699
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3788
3789
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3798
3799
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3888
3889
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3898
3899
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3988
3989
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3998
3999
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4088
4089
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4098
4099
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4188
4189
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4198
4199
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4288
4289
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4298
4299
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4388
4389
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4398
4399
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4488
4489
4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4498
4499
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508
4509
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4539
4540
4541
4542
4543
4544
4545
4546
4547

```

```

2592
2593     2. Choose the algorithm that best matches the server instructions.
2594     Remember, your choice should be mainly based on "Full Name", "Key
2595     idea" entries.
2596     3. Return the chosen algorithm as Algorithm Name: ....
2597     4. Include <end> to end the conversation.
2598     """
2599     return system_prompt
2600
2601
2602     def FL_trainer_prompt(project_directory, selected_algorithm):
2603         system_prompt = f"""
2604             You are a trainer agent that performs federated learning with
2605             selected clients using the chosen algorithm: {selected_algorithm}
2606             You have access to the tools:
2607             run_federated_method: Runs the specified federated learning
2608             Use run_federated_method to run the specific federated learning
2609             algorithm: {selected_algorithm} and report results.
2610             """
2611         return system_prompt
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645

```

Table 3: Summary of Specialized Agents and Their Responsibilities in Federated Learning Workflow

Agent	Agent Name	Role Description	Phase
S_1	Server Agent for Task Interpretation	Parses user instructions to extract task and modality requirements; broadcasts the requirement to all client agents to begin dataset selection.	Phase 1: Client Selection
C_1	Client Selector Agent	Evaluates dataset metadata to identify relevant datasets for the task based on textual descriptions in a JSON file; responds with matched datasets or "no dataset".	Phase 1: Client Selection
S_2	Server Agent for Client Approval	Reviews responses from clients; approves those with relevant datasets for training or excludes irrelevant ones.	Phase 1: Client Selection
C_2	Data Pre-processor Agent	Organizes dataset into class-wise subfolders, removes non-image files, and performs data cleaning (e.g., de-duplication, noise filtering, off-topic detection).	Phase 2: Data Preparation
C_3	Task-conditioned Label Harmonizer Agent	Reorganizes client label spaces into harmonized schema by mapping fine-grained classes to broader target labels (e.g., malignant, benign).	Phase 3: Label Harmonization
S_3	FL Algorithm Selector Agent	Selects the most appropriate federated learning algorithm based on the user's task by analyzing algorithm metadata.	Phase 4: FL Algorithm Selection
S_4	Trainer Agent	Executes the federated learning training using the chosen algorithm and the approved client datasets.	Phase 4: Federated Training

C TASKS AND ALGORITHMS IN FEDAGENTBENCH FRAMEWORK

C.1 DATASET DETAILS

To enable systematic benchmarking across a broad range of real-world clinical scenarios, FedAgentBench includes 201 publicly available datasets spanning six major medical imaging modalities: Dermatology (25 datasets), Ultrasound (33), Fundus (63), X-Ray (32), MRI (28), and Histopathology (20). These datasets comprise both 2D and 3D imaging formats and cover a wide array of task types, including classification (e.g., tumor detection, cancer subtype identification), grading/staging (e.g.,

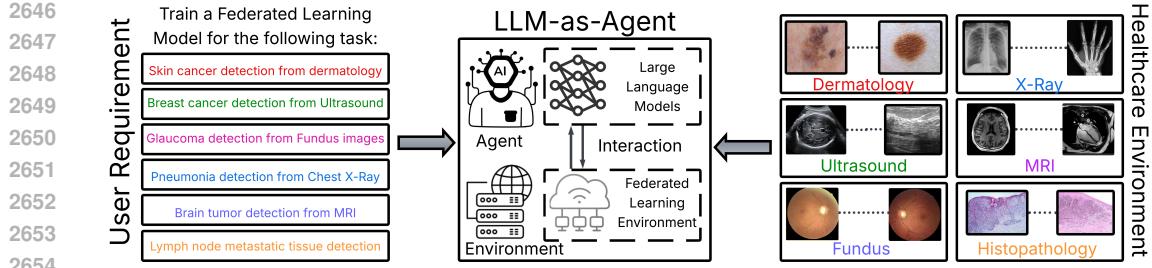


Figure 9: Sample tasks and datasets in FedAgentBench

diabetic retinopathy, cancer severity), segmentation (e.g., lesion, tumor, or stroke localization), object detection, regression, image reconstruction, and registration.

Each client in FedAgentBench is simulated by grouping one or more of these datasets, thereby reflecting the diversity and data heterogeneity found in real-world healthcare settings. For each client, a datacard is constructed, compiling metadata sourced from the original dataset publication, repository, or project website. This metadata includes information on imaging modality, data dimensionality, task type, class schema, and clinically relevant attributes, ensuring traceability and reproducibility.

In the following subsections, we provide a detailed breakdown of the dataset description for each imaging modality.

DERMATOLOGY:

The dermatology dataset collection curated for this benchmark represents one of the most comprehensive and heterogeneous sets assembled for machine learning research in skin disease analysis. Spanning over 25 datasets, the collection includes both photographic and dermoscopic images, structured tabular data, and multi-modal formats. The classification tasks range from binary cancer detection (e.g., benign vs. malignant in *ISIC2020*, *Mednode*) to fine-grained multi-class diagnosis involving over twenty conditions (e.g., *Dermnet*, *Derma7PT*, *skinL2_dataset*). Several datasets such as *DDI_skin_dataset* and *fitzpatrick17k* are designed to ensure skin tone diversity, while others like *Monkeypox_Skin_Image_Dataset* and *skin-infection-disease-dataset* address emerging and infectious conditions. Additionally, datasets like *PH2Dataset*, *ISIC2016–2024*, and *Dermis* support segmentation and localization, enabling both classification and pixel-wise lesion analysis. This diversity reflects a realistic, clinically relevant spectrum of dermatological challenges, and is particularly well-suited for benchmarking federated learning agents under varying input types, diagnostic complexity, and data distributions. The code snippets for dermatology dataset description file can be found in Listing 6. The description of each dataset is summed up below:

1. augmented_skin_condition_dataset_kaggle. The *augmented_skin_condition_dataset_kaggle* dataset (aug) is designed for multi-class skin disease classification. It contains photographic images of six dermatological conditions: Acne, Carcinoma, Eczema, Keratosis, Milia, and Rosacea, supporting automated detection and differentiation of common skin ailments.

2. DDI_skin_dataset. The *DDI_skin_dataset* (Daneshjou et al., 2022) is a skin cancer classification resource with strong representation of diverse skin tones. Each image is annotated as benign or malignant, enabling the development of robust melanoma and non-melanoma skin cancer detection algorithms for varied populations.

3. Derma7PT. *Derma7PT* (Kawahara et al., 2018) is a multi-class skin disease classification dataset, annotated with ten distinct diagnostic categories: basal cell carcinoma, nevus, dermatofibroma, lentigo, melanoma, melanoma metastasis, melanosis, miscellaneous, seborrheic keratosis, and vascular lesion. It is suitable for fine-grained disease discrimination in clinical dermatology.

4. Dermatology_tabular dataset. The *Dermatology_tabular* (Der, a) dataset provides structured clinical features for diagnosing various skin diseases. It is intended for the development and bench-

2700 marking of machine learning models using tabular (non-image) data for dermatological decision
 2701 support.
 2702

2703 **5. Dermis.** *Dermis* (Der, b) is a dual-purpose dataset supporting both skin lesion classification
 2704 (benign vs malignant) and lesion segmentation. It is suitable for the development of algorithms
 2705 targeting melanoma recognition and precise lesion boundary detection.
 2706

2707 **6. Dermnet.** *Dermnet* (Der, c) is a broad dermatology image dataset encompassing 23 disease
 2708 categories, ranging from inflammatory conditions (e.g., eczema, psoriasis) to infectious (bacterial,
 2709 viral, fungal), neoplastic (melanoma, carcinoma), and other rare skin diseases. It is valuable for
 2710 comprehensive multi-class skin disease classification.
 2711

2712 **7. Dermquest.** *Dermquest* (Der, d) offers images for both classification (benign vs malignant) and
 2713 segmentation of skin lesions, supporting research in melanoma detection as well as pixel-wise lesion
 2714 analysis.
 2715

2716 **8. Fitzpatrick17k.** The *fitzpatrick17k* (Groh et al., 2021) dataset features a wide range of derma-
 2717 tological disease images, annotated with three high-level categories: non-neoplastic, benign, and
 2718 malignant. Its diverse cases make it well suited for studying skin cancer classification across various
 2719 skin tones.
 2720

2721 **9. ISIC2018_HAM10000.** The *ISIC2018_HAM10000* (Codella et al., 2019) dataset is a stan-
 2722 dard benchmark for skin lesion diagnosis and segmentation, including cases such as melanocytic
 2723 nevus, benign keratosis, melanoma, basal cell carcinoma, actinic keratosis, vascular lesions, and
 2724 dermatofibroma. It is used for both classification and lesion segmentation.
 2725

2726 **10. ISIC_2016.** *ISIC_2016* (Gutman et al., 2016) supports binary classification (benign vs ma-
 2727 lignant) and lesion segmentation for skin cancer detection, with a focus on melanoma diagnosis in
 2728 clinical dermoscopic images.
 2729

2730 **11. ISIC_2017.** *ISIC_2017* (Berseth, 2017) targets the detection and segmentation of melanoma
 2731 and seborrheic keratosis in dermoscopic images, supporting both binary and multi-label skin cancer
 2732 classification tasks.
 2733

2734 **12. ISIC_2019.** The *ISIC_2019* (Combalia et al., 2019) dataset offers an expanded benchmark for
 2735 skin disease classification, with images labeled for nine conditions including melanoma, nevus, basal
 2736 cell carcinoma, actinic keratosis, and others, facilitating studies in multi-class lesion recognition.
 2737

2738 **13. ISIC_2020.** *ISIC_2020* (ISI, a) is a binary skin lesion classification dataset, primarily focused
 2739 on discriminating benign from malignant lesions in dermoscopic images for melanoma screening.
 2740

2741 **14. ISIC_2024.** The *ISIC_2024* (ISI, b) dataset continues the ISIC challenge series with an updated
 2742 collection focused on binary melanoma (benign vs malignant) classification for automated skin cancer
 2743 diagnosis.
 2744

2745 **15. Mednode.** *Mednode* (MED) is a binary classification dataset distinguishing between melanoma
 2746 and nevus, intended for the development and validation of melanoma detection models.
 2747

2748 **16. Monkeypox_Skin_Image_Dataset.** The *Monkeypox_Skin_Image_Dataset* (Mon) supports
 2749 image-based classification of viral skin diseases, including Monkeypox, Chickenpox, Measles, and
 2750 Normal skin, for research on differential diagnosis of infectious exanthems.
 2751

2752 **17. PAD_UFES_20.** *PAD_UFES_20* (Pacheco et al., 2020) provides images and diagnostic la-
 2753 bels for six skin tumor types: melanoma, melanocytic nevus, basal cell carcinoma, actinic kerato-
 2754 sis/Bowen’s disease, seborrheic keratosis, and squamous cell carcinoma, supporting both single- and
 2755 multi-class lesion classification.
 2756

2754 **18. PH2Dataset.** The *PH2Dataset* (PH2) contains dermoscopic images and expert-annotated
 2755 segmentation masks for three classes: common nevus, atypical nevus, and melanoma, making it
 2756 suitable for both lesion segmentation and classification.
 2757

2758 **19. scin_dataset.** *scin_dataset* (Ward et al., 2024) is a multi-class classification dataset including a
 2759 range of common skin diseases, such as acne, pigmentary problems, nail disorders, hair loss, and
 2760 others, for developing comprehensive skin disease classifiers.
 2761

2762 **20. skin_disease_3_class.** The *skin_disease_3_class* dataset comprises images for classifying three
 2763 skin diseases: acne, atopic dermatitis, and basal cell carcinoma.
 2764

2765 **21. skin_disease_classification_kaggle.** *skin_disease_classification_kaggle* (ski, a) is a small
 2766 dataset for multi-class classification of acne, eye bags, and redness, designed for image-based
 2767 diagnosis of common cosmetic and inflammatory skin conditions.
 2768

2769 **22. skin_disease_kaggle_dataset.** The *skin_disease_kaggle_dataset* supports multi-class skin
 2770 disease classification for ten clinically relevant categories, including atopic dermatitis, basal cell
 2771 carcinoma, eczema, melanoma, nevi, psoriasis, seborrheic keratosis, and infectious diseases.
 2772

2773 **23. Skin Disease_Robo.** *Skin Disease_Robo* is a skin disease dataset for both image classification
 2774 and object detection. It provides bounding box annotations for ten skin disease classes, including
 2775 acne, atopic dermatitis, eczema, leprosy, psoriasis, ringworm, and warts.
 2776

2777 **24. skin-infection-disease-dataset.** The *skin-infection-disease-dataset* (ski, b) focuses on the
 2778 classification of eight infectious skin diseases, covering bacterial, fungal, parasitic, and viral infections
 2779 such as cellulitis, impetigo, athlete’s foot, ringworm, cutaneous larva migrans, chickenpox, and
 2780 shingles.
 2781

2782 **25. skinL2_dataset.** The *skinL2_dataset* (de Faria et al., 2019) is a skin cancer classification
 2783 resource annotated for eight disease classes, including basal cell carcinoma, dermatofibroma, hemangioma,
 2784 melanoma, nevus, psoriasis, seborrheic keratosis, and others, facilitating both melanoma and
 2785 non-melanoma skin lesion research.
 2786

ULTRASOUND:

2787 The ultrasound dataset collection constitutes a diverse and representative corpus of ultrasound
 2788 images. Spanning over 33 datasets, this collection captures the breadth of clinical applications
 2789 across multiple anatomical regions (e.g., breast, fetal brain, liver, thyroid, heart, vascular system,
 2790 musculoskeletal structures), imaging modalities (e.g., B-mode, Doppler, color flow), and task types
 2791 (e.g., classification, segmentation, super-resolution, registration). Classification challenges range
 2792 from binary diagnostic tasks such as benign vs. malignant lesion detection (e.g., BUSI, Mendeley,
 2793 BUET BUSD) to multi-class pathological condition analysis (e.g., PCOS detection, fetal health
 2794 classification). Several datasets, such as FALLMUD and fetal head US, are curated to support
 2795 precise biometric measurements and fetal growth monitoring, while others such as CAMUS and leg
 2796 segmentation datasets are tailored for structure delineation critical in cardiology and musculoskeletal
 2797 rehabilitation, respectively. The inclusion of multimodal and cross-domain datasets—such as MUS-V
 2798 (vascular segmentation from Doppler and B-mode), CT2US (CT-to-ultrasound adaptation), and Ultra
 2799 LR-HR (super-resolution) further enhances the heterogeneity of input formats and computational
 2800 tasks. In addition, the dataset collection includes rare or emerging clinical tasks such as dermatologic
 2801 ultrasound, liver fibrosis staging, and hemangioma classification, reflecting real-world diagnostic
 2802 diversity. This rich variation of organs, pathologies, modalities, and task complexities makes the
 2803 benchmark exceptionally well-suited for evaluating federated learning agents under diverse diagnostic
 2804 conditions, cross-institutional generalization scenarios, and clinically realistic constraints.
 2805

2806 **1. Breast Ultrasound Images (BUSI):** This dataset (BUS, b) is used for images of breast tumors
 2807 annotated as benign, malignant, or normal. Specifically, it aims to detect and classify breast tumors
 2808 into benign, malignant, or normal categories, and delineate the exact tumor boundaries in ultrasound
 2809 images.
 2810

2808 **2. B-mode fatty liver US images:** This dataset (Byra et al., 2018) is used for ultrasound images
 2809 used to classify liver steatosis severity. Specifically, it aims to assess and classify the degree of fatty
 2810 liver disease (hepatic steatosis) using grayscale B-mode ultrasound scans.

2811
 2812 **3. Fetal health classification:** This dataset (Fet, b) is used for ultrasound data related to fetal health
 2813 status. Specifically, it aims to evaluate fetal condition based on cardiotocographic or ultrasound
 2814 signals to classify into normal, suspected, or pathological health status.

2815
 2816 **4. Robotic handheld lumbar spine US:** This dataset (Rob) is used for ultrasound images of
 2817 lumbar spine captured with robotic devices. Specifically, it aims to identify and segment vertebrae
 2818 and surrounding spinal anatomy from ultrasound images acquired by a robotic handheld device for
 2819 navigation.

2820
 2821 **5. BUS-UCLM:** This dataset (BUS, a) is used for breast ultrasound dataset from uclm annotated
 2822 for tumors. Specifically, it aims to differentiate between benign and malignant breast lesions and
 2823 segment the tumor region for further morphological analysis.

2824
 2825 **6. Regensburg pediatric appendicitis:** This dataset (Reg) is used for ultrasound images of
 2826 pediatric patients for appendicitis diagnosis. Specifically, it aims to distinguish between pediatric
 2827 patients with and without appendicitis based on ultrasound scans of the abdomen.

2828
 2829 **7. Breast Ultrasound Images:** This dataset (Bre, b) aims to support breast cancer diagnosis by
 2830 classifying tumors and extracting the region of interest (ROI) for clinical examination.

2831
 2832 **8. BUS-UC:** This dataset (Al-Dhabayani et al., 2020) is used for breast ultrasound dataset from
 2833 university of california. Specifically, it aims to classify ultrasound-detected breast abnormalities and
 2834 perform segmentation to assist in diagnostic workflows.

2835
 2836 **9. Fetal head US dataset:** This dataset (Fet, a) is used for images focused on fetal head for
 2837 biometry (e.g., hc, bpd). Specifically, it aims to extract biometric measurements such as biparietal
 2838 diameter (BPD) and head circumference (HC) through segmentation of the fetal head.

2839
 2840 **10. Carotid Ultrasound Images:** This dataset (Car, a) is used for ultrasound images of carotid
 2841 arteries, with plaque annotations. Specifically, it aims to detect carotid artery plaques and measure
 2842 intima-media thickness (IMT) to evaluate cardiovascular risk.

2843
 2844 **11. Ultrasound breast images (for cancer):** This dataset is used for breast cancer detection.
 2845 Specifically, it aims to classify breast lesions as benign or malignant in 2D ultrasound scans for early
 2846 cancer detection.

2847
 2848 **12. 3D MRI Ultrasound brain images:** This dataset (3D) is used for magnetic resonance elastog-
 2849 raphy and ultrasound for brain imaging. Specifically, it aims to analyze brain stiffness and segment
 2850 relevant anatomical regions in elastography-enhanced 3D ultrasound volumes.

2851
 2852 **13. CAMUS Human Heart:** This dataset (CAM) is used for 2D echocardiographic sequences with
 2853 lv, myocardium, and la labels. Specifically, it aims to segment key cardiac structures such as the left
 2854 ventricle (LV), myocardium, and left atrium from 2D echocardiography sequences.

2855
 2856 **14. CT2US for Kidney Seg:** This dataset (CT2) is used for CT-derived kidney masks mapped
 2857 to US domain. Specifically, it aims to leverage CT-derived kidney masks to train ultrasound-based
 2858 models for accurate kidney segmentation under domain adaptation.

2859
 2860 **15. Breast Cancer Image Dataset:** This dataset (Bre, a) is used for breast cancer detection.
 2861 Specifically, it aims to differentiate benign and malignant breast lesions to assist in non-invasive
 2862 cancer diagnosis.

2862 **16. DDTI: Thyroid US Images:** This dataset (DDT) is used for digital database for thyroid
 2863 imaging with nodule annotations. Specifically, it aims to detect and classify thyroid nodules and
 2864 delineate their contours to support risk stratification and clinical reporting.
 2865

2866 **17. Thyroid Ultrasound:** This dataset (Thy) is used for thyroid nodule dataset. Specifically, it
 2867 aims to perform classification and detailed boundary segmentation of thyroid nodules from grayscale
 2868 ultrasound scans.
 2869

2870 **18. Multimodal Breast US Dataset (US3M):** This dataset (US3) is used for multimodal dataset
 2871 with us, mri, mammo for breast lesions. Specifically, it aims to fuse features from mammography,
 2872 MRI, and ultrasound to enhance breast tumor classification using multimodal representations.
 2873

2874 **19. Liver histopathology (Fibrosis):** This dataset (Liv) is used for ultrasound images labeled with
 2875 fibrosis grades based on biopsy. Specifically, it aims to grade liver fibrosis severity from ultrasound
 2876 images based on corresponding histopathological findings from biopsy.
 2877

2878 **20. Prostate MRI and Ultrasound:** This dataset (pro, b) is used for prostate cancer detection
 2879 using mri and us fusion. Specifically, it aims to segment the prostate gland and align ultrasound scans
 2880 with MRI images for guided prostate biopsy or treatment planning.
 2881

2882 **21. Carotid artery US & Color Doppler** This dataset (Car, b) is used for detecting stenosis
 2883 and plaque buildup in the carotid arteries. It typically includes segmentation of the vessel wall
 2884 and atherosclerotic plaque, along with classification of stenosis severity using Doppler blood flow
 2885 analysis.
 2886

2887 **22. PCOS Detection using Ultrasound Images** This dataset (PCO) involves classifying ovarian
 2888 ultrasound images to detect Polycystic Ovary Syndrome (PCOS). Features such as ovarian volume,
 2889 follicle count, and echogenicity are commonly used for diagnosis.
 2890

2891 **23. Ultra LR-HR Ultrasound Dataset** An ultrasound dataset (ult, a) used for super-resolution
 2892 tasks, where low-resolution ultrasound images are enhanced or reconstructed into high-resolution
 2893 versions.
 2894

2895 **24. MUS-V (Multimodal Ultrasound Vascular Segmentation)** This dataset (mul) integrates
 2896 multiple ultrasound modalities such as B-mode and Doppler to improve the accuracy of vascular
 2897 segmentation tasks.
 2898

2899 **25. BUET BUSD** Developed by the Bangladesh University of Engineering and Technology (BUE),
 2900 this breast ultrasound dataset is used for both classification and segmentation of lesions.
 2901

2902 **26. Dermatologic Ultrasound Images** An emerging application of ultrasound for skin lesions
 2903 (der). This dataset is used for classifying dermatological conditions such as melanomas, cysts, or
 2904 benign tumors.
 2905

2906 **27. FHMS Ultrasound Dataset** This is a fetal head ultrasound dataset (fhm).
 2907

2908 **28. Mendeley Breast Ultrasound Dataset** A publicly available dataset (men) containing 780
 2909 images labeled as benign, malignant, or normal. It is frequently used for breast lesion classification.
 2910

2911 **29. FALLMUD** Fetal Abdomen and Longitudinal Liver Measurement in Ultrasound Dataset (fal)
 2912 is used for segmentation of the fetal abdomen and liver, important for fetal growth monitoring.
 2913

2914 **30. Leg Segmentation – Ultrasound** This dataset (leg) focuses on segmenting muscles, tendons,
 2915 and fasciae in ultrasound images of the lower limbs. It has applications in physical therapy and sports
 medicine.
 2916

2916 **31. Fetal Ultrasound Brain** A dataset of fetal brain ultrasounds (fet), commonly used for
 2917 segmenting brain structures such as the lateral ventricles and midline. It supports fetal development
 2918 tracking.
 2919

2920 **32. Ultrasound Image Set of Hemangiomas** This dataset includes ultrasound images of heman-
 2921 giomas, which are benign vascular tumors. It is used for classifying these from other types of soft
 2922 tissue lesions.
 2923

2924 **33. Ultrasound Nerve Segmentation** This dataset (ult, b) comprises ultrasound images for
 2925 identifying nerve structures of the neck. This would lead to improvement in catheter placement and
 2926 contribute to reduction in post-surgical pain.
 2927

2928 **X-RAY:**
 2929

2930 The X-ray dataset collection in FedAgentBench represents a highly diverse benchmark suite, en-
 2931 compassing 32 datasets across multiple diagnostic and anatomical categories. It includes chest,
 2932 bone, knee, dental, and vascular imaging modalities, with tasks ranging from binary classification
 2933 (e.g., pneumonia vs. normal in *pneumonia*, COVID-19 vs. normal in *cov_19* and *cov19_normal*)
 2934 to complex multi-class and object detection tasks (e.g., *xray_17_diseases*, *8_object_detection*, and
 2935 *RSNA-breast-cancer-detection*). Several datasets offer bounding box or pixel-wise segmentation
 2936 annotations (*NIH_bbox*, *lung_segmentation*, *PAX-Ray++*), while others contain structured metadata
 2937 (e.g., *spr_age_gender*, *knee*, *RANZCR*), enabling multi-modal reasoning and demographic prediction.
 2938 This collection also includes modality-bridging datasets like *HBFMID* that pair X-ray and MRI scans,
 2939 and datasets that focus on disease-specific localization such as *humerus_fractures*, *HeelBone*, and
 2940 *FracAtlas*. Collectively, the X-ray corpus provides a robust foundation for evaluating LLM agents
 2941 on a wide range of radiological tasks—spanning classification, segmentation, detection, and clinical
 2942 interpretation under realistic federated learning constraints. The exact dataset descriptions prepared
 2943 for the client selection agents are provided in Listing 7 and summarized below:
 2944

2945 **1. cov_19.** The *cov_19* dataset (Rahman, 2020) comprises chest X-ray images collected by an
 2946 international team of researchers, featuring COVID-19 positive cases alongside normal and viral
 2947 pneumonia images. Initially released with 219 COVID-19, 1,341 normal, and 1,345 viral pneumonia
 2948 images, the dataset has since expanded to include 3,616 COVID-19 cases, 10,192 normal cases, 6,012
 2949 lung opacity (non-COVID lung infection) cases, and 1,345 viral pneumonia cases. Each update has
 2950 added more images and corresponding lung masks. Data sourcing and ongoing updates make this
 2951 dataset a valuable resource for developing robust models for COVID-19 and other lung diseases.
 2952

2953 **2. bone_frac.** The *bone_frac* dataset (Rodrigo, 2022) includes X-ray images of fractured and
 2954 non-fractured bones across various anatomical regions, such as the lower and upper limbs, lumbar
 2955 spine, hips, and knees. The images are divided into train, test, and validation sets, each containing
 2956 both classes, making the dataset suitable for training and evaluating bone fracture detection and
 2957 classification algorithms.
 2958

2959 **3. chest_tuberculosis_segmentation.** The *chest_tuberculosis_segmentation* dataset (Tapendu,
 2960 2023a) consists of 704 chest X-ray images sourced from the Montgomery County Chest X-ray
 2961 Database (USA) and the Shenzhen Chest X-ray Database (China). It includes tuberculosis-positive
 2962 and normal images, accompanied by lung segmentation masks and clinical metadata (e.g., age, gender,
 2963 county of origin). The combination of images and annotations makes it suitable for tuberculosis
 2964 detection, segmentation, and broader deep learning tasks in medical imaging.
 2965

2966 **4. xray_17_diseases.** The *xray_17_diseases* dataset (TrainingDataPro, 2023) offers chest X-ray
 2967 images in both .jpg and .dcm formats, labeled for a diverse set of thoracic diseases, including abscess,
 2968 ARDS, atelectasis, atherosclerosis, cardiomegaly, emphysema, fractures, pneumonia, tuberculosis,
 2969 and more. The dataset supports research in neurology, radiology, and oncology, enabling the
 development and evaluation of models for automated disease detection, diagnosis, and classification.
 2970

2970 **5. spr_age_gender.** The *SPR Age and Gender* dataset (Kitamura, 2022a) contains X-ray images
 2971 in .png format with accompanying CSV files specifying patient age and gender. It is designed for
 2972 research on patient demographic prediction from radiographic data.
 2973

2974 **6. unifesp.** The *UNIFESP X-Ray Body Part Classification* dataset (Kitamura, 2022b) comprises
 2975 2,481 DICOM-format X-ray images annotated by radiology residents. The dataset covers 20 anatomical
 2976 body parts (plus an “other” category), with categorical labels assigned to each image, supporting
 2977 multi-label classification tasks and body part recognition in medical imaging.
 2978

2979 **7. knee.** This dataset (Orvile, 2023d) contains 1,650 high-quality digital X-ray images of the
 2980 knee, manually annotated by medical experts using the Kellgren and Lawrence grading system for
 2981 osteoarthritis severity. The images are 8-bit grayscale and are accompanied by metadata and cartilage
 2982 region annotations, facilitating research in automated knee osteoarthritis detection and grading.
 2983

2984 **8. c19_radiograph.** The *c19_radiograph* dataset (Viradiya, 2023) is a comprehensive chest X-ray
 2985 collection curated by a team from Qatar University and the University of Dhaka, with COVID-19,
 2986 normal, lung opacity, and viral pneumonia cases. The database is built from multiple public and
 2987 hospital sources and contains extensive clinical labels and patient metadata, enabling detailed studies
 2988 of COVID-19 pneumonia and related conditions.
 2989

2990 **9. simple_vs_community.** This bone fracture dataset (Orvile, 2023b) is structured to distinguish
 2991 between simple and comminuted fractures, comprising over 7,500 images for simple fractures and
 2992 more than 8,500 for comminuted fractures. It combines hospital records and web-sourced images,
 2993 and includes extensive data augmentation, providing a challenging dataset for fracture classification
 2994 and segmentation tasks.
 2995

2996 **10. nih_bbox.** The *NIH Chest X-ray* dataset (Hodeb, 2023) consists of 112,120 images from
 2997 30,805 patients, each labeled for thoracic diseases using text-mined radiology reports. The dataset
 2998 features bounding box annotations for localization, supports weakly-supervised learning, and includes
 2999 metadata on disease classes, patient demographics, and imaging protocols.
 3000

3001 **11. bone_break.** The *bone_break* dataset (Darabi, 2023) focuses on the classification of various
 3002 bone fracture types using X-ray images. It encompasses multiple fracture classes, such as avulsion,
 3003 comminuted, fracture-dislocations, greenstick, hairline, impacted, longitudinal, oblique, pathological,
 3004 and spiral fractures, supporting the development of automated fracture classification systems.
 3005

3006 **12. cov19_normal.** This balanced dataset (Tejas, 2022) contains 800 high-quality chest X-ray
 3007 images, equally divided between COVID-19 positive and normal cases (400 each). The curated and
 3008 balanced nature makes it ideal for deep learning studies on COVID-19 detection.
 3009

3010 **13. dental.** The *dental* dataset (IMT Kaggle Team, 2023) consists of dental radiographs, enabling
 3011 the evaluation of hard and soft tissue changes, jawbone development in children, and the detection of
 3012 injuries in facial and oral structures. It is suitable for a range of dental diagnostic research tasks.
 3013

3014 **14. bone_frac_small.** A focused dataset (Orvile, 2023a) for bone fracture classification and
 3015 localization in tibia and fibula bones, *bone_frac_small* features X-ray images in PNG format. Some
 3016 images have been validated by medical experts at the University of Gondar, Ethiopia. The dataset
 3017 includes enhanced and augmented images for robust model development.
 3018

3019 **15. knee_osteoporosis.** Sourced from Mendeley Data, the *knee_osteoporosis* dataset (Gobara,
 3020 2023b) contains X-rays categorized into three classes: normal, osteopenia, and osteoporosis. It is
 3021 intended for studies on bone density assessment and osteoporosis detection.
 3022

3023 **16. RNSA_pneumonia.** A pre-processed version of the RSNA Pneumonia Detection Challenge
 3024 dataset, *RNSA_pneumonia* (Tapendu, 2023b) includes PNG images and mask-based bounding box
 3025 annotations. Associated metadata, such as patient information and bounding box coordinates, is
 3026 provided in CSV format for easy integration into machine learning pipelines.

3024 **17. 8_object_detection.** The *Chest X-ray 8 Subset* (Spiritan1, 2023) is tailored for object detection
 3025 in thoracic diseases, containing 790 images with 984 bounding boxes. Annotations are available in
 3026 YOLO and Pascal VOC formats, and the dataset includes 14 thoracic disease classes, facilitating the
 3027 development of object detection models in medical imaging.

3028
 3029 **18. HBFMID.** The *Human Bone Fractures Multi-modal Image Dataset* (HBFMID) (Orville, 2023c)
 3030 includes 1,539 annotated images (X-ray and MRI) covering fractures at multiple anatomical sites.
 3031 The dataset is divided into training, validation, and testing sets and has undergone preprocessing (auto-
 3032 orientation, resizing, contrast adjustments), supporting research in multi-modal fracture diagnosis.

3033
 3034 **19. FracAtlas.** *FracAtlas* (Gupta, 2023) comprises over 14,000 X-ray scans collected from three
 3035 major hospitals in Bangladesh, with 4,083 images manually annotated for bone fracture classification,
 3036 localization, and segmentation. Annotations were conducted by expert radiologists and validated by a
 3037 medical officer, providing a high-quality benchmark for fracture analysis.

3038
 3039 **20. pneumonia.** The *pneumonia* dataset (Mooney, 2018) contains 5,863 chest X-ray images
 3040 (anterior-posterior) of pediatric patients, labeled as either pneumonia or normal. Images underwent
 3041 strict quality control and multi-expert grading, making the dataset reliable for training AI systems in
 3042 pneumonia detection.

3043
 3044 **21. pax_ray.** The *PAX-Ray++* dataset (Seibold, 2023) contains 7,377 chest radiographs (frontal
 3045 and lateral views), with pseudo-labeled annotations for anatomical segmentation generated from
 3046 projected thorax CT scans. The dataset is designed for segmentation tasks in chest X-ray analysis.

3047
 3048 **22. lung_segmentation.** This dataset (Beosup, 2023) consists of over 500 X-ray scans labeled by
 3049 radiologists, supporting machine learning research in lung region segmentation.

3050
 3051 **23. shadow.** The *shadow* dataset (Hmchuong, 2023) includes normal and bone-suppressed chest
 3052 X-ray images, along with augmented samples. It is intended for research on bone shadow suppression
 3053 to aid in lung disease diagnosis.

3054
 3055 **24. Angiography.** The *ARCADE* dataset (Manaenkov, 2023) features 3,000 X-ray coronary an-
 3056 giography frames with expert annotations for vessel segmentation, SYNTAX scoring, and stenosis
 3057 detection. It is organized by task and includes cross-validated annotations, providing a rich resource
 3058 for AI research in coronary artery disease diagnostics.

3059
 3060 **25. dental_panoramic.** This panoramic dental radiograph dataset (Lokisilvres, 2023) includes
 3061 segmentation masks for 31 dental disease classes, such as caries, crowns, implants, bone loss,
 3062 fractures, and more. It is intended for comprehensive dental disease detection and segmentation
 3063 research.

3064
 3065 **26. ALHI.** The *ALHI* dataset (Rahman, 2022) is a curated collection of 200 hip implant X-ray
 3066 images from various medical sources, annotated and validated by orthopedic and clinical experts.
 3067 The dataset includes images with diverse implant types and clinical conditions, supporting research
 3068 on hip implant assessment.

3069
 3070 **27. humerus_fractures.** The *humerus_fractures* dataset (Paspuel, 2024) contains X-ray images
 3071 depicting both fractured and non-fractured humeri, supporting automated diagnosis of humerus
 3072 fractures through deep learning.

3073
 3074 **28. multiclass_knee_osteoporosis.** This dataset (Gobara, 2023a) offers X-ray images and patient
 3075 records classified into normal, osteopenia, and osteoporosis categories, facilitating the automated
 3076 diagnosis and classification of knee osteoporosis.

3077
 3078 **29. rsna-breast-cancer-detection.** The *RSNA Breast Cancer Detection* dataset (Thakur, 2024)
 3079 provides breast X-ray image regions of interest (ROIs) in PNG format, without labels, for studies on
 3080 automated detection in breast imaging.

3078 **30. RANZCR.** The *RANZCR* dataset (RANZCR, 2021) is intended for detecting the presence and
 3079 position of catheters and lines on chest X-rays. It contains image IDs, binary labels for multiple types
 3080 of catheters, and patient identifiers, along with associated CSV metadata.
 3081

3082 **31. FractureFusion.** *FractureFusion* (Dutta, 2023) is a diverse dataset capturing a wide variety of
 3083 bone fracture cases, including avulsion, comminuted, greenstick, and spiral fractures, suitable for
 3084 developing comprehensive fracture classification models.
 3085

3086 **32. HeelBone.** The *Heel Bone X-Ray* dataset (Taher, 2023) comprises 3,956 foot X-rays labeled for
 3087 normal, heel spur, and severe heel spur complications. Images were sourced from Kirkuk General
 3088 Hospital and cross-verified by orthopedic and radiology specialists, supporting disease classification
 3089 in foot imaging.
 3090

3091 HISTOPATHOLOGY:

3092 The histopathology dataset collection in FedAgentBench covers a wide range of diseases and task
 3093 types, making it a comprehensive benchmark for evaluating LLM agents in digital pathology. It spans
 3094 various cancer types, including breast (e.g., *breast_histo*, *BreaKHis_400X*, *BreCaHAD*), ovarian
 3095 (*ovarian_cancer*), gastric (*gastric_cancer*), kidney (*kmc_kidney*), melanoma, and nasopharyngeal
 3096 carcinoma (*NPC-88k-Public*). The datasets support multiple learning paradigms such as binary and
 3097 multi-class classification (*lung_and_colon*, *EBHI*), segmentation (*MonuSeg*, *PanNuke*), detection of
 3098 mitotic figures (*ULMS*), and multimodal image-to-text learning (*histo-img-text*). Some datasets, like
 3099 *choledoch*, incorporate hyperspectral imaging, while others like *CellNet* aggregate thousands of high-
 3100 resolution images across organ types, facilitating generalization studies. Fine-grained annotations by
 3101 expert pathologists (e.g., in *BreCaHAD*, *NPC-88k-Public*, *MonuSeg*) add clinical reliability. Together,
 3102 these datasets reflect a realistic landscape of digital histopathology rich in diagnostic complexity,
 3103 varied in modality and scale, and suitable for evaluating both general-purpose and specialized LLM
 3104 agents in federated clinical settings. The exact dataset descriptions for each file are available in
 3105 Listing 8 and summarized as follows:
 3106

3107 **1. breast_histo.** The *Breast Histopathology Images* dataset (Mooney, 2024) focuses on Invasive
 3108 Ductal Carcinoma (IDC), the most common breast cancer subtype. The original dataset comprises
 3109 162 whole mount slides scanned at 40x magnification, from which 277,524 patches of size 50×50
 3110 were extracted (198,738 IDC negative and 78,786 IDC positive). Patch filenames encode patient ID,
 3111 spatial coordinates, and IDC class (0 for non-IDC, 1 for IDC). Only images are provided, with no
 3112 additional labels.
 3113

3114 **2. BreaKHis_400X.** The *BreaKHis_400X* dataset (Forderation, 2024) is derived from the BreaKHis
 3115 database, which contains microscopic biopsy images of benign and malignant breast tumors. This
 3116 subset includes images acquired at 400x optical zoom, with training and test data stored in separate
 3117 folders. Images only are provided; no labels are included.
 3118

3119 **3. lung_and_colon.** The *Lung and Colon Cancer Histopathological Images* dataset (MVD, 2024a)
 3120 contains 25,000 JPEG images of size 768×768 pixels, covering five classes: lung benign tissue, lung
 3121 adenocarcinoma, lung squamous cell carcinoma, colon adenocarcinoma, and colon benign tissue.
 3122 Images were generated from HIPAA-compliant and validated original samples (750 lung and 500
 3123 colon images) and augmented using the Augmentor package to create a balanced dataset of 5,000
 images per class.
 3124

3125 **4. gastric_cancer.** The *Gastric Cancer Histopathology Tissue Image Dataset* (GCHTID) (Orville,
 3126 2024) comprises 31,096 non-overlapping images (224×224 pixels), extracted from H&E-stained
 3127 pathological slides from Harbin Medical University Cancer Hospital. Images are categorized into
 3128 eight tissue types, including adipose, background, debris, lymphocytes, mucus, smooth muscle,
 3129 normal colon mucosa, cancer-associated stroma, and tumor, enabling research on the tumor microen-
 3130 vironment in gastric cancer.
 3131

3132 **5. gastro_cancer_msi_vs_mss.** The *Gastrointestinal Cancer MSI MSS Prediction* dataset (Justin,
 3133 2024) contains histological images for the classification of microsatellite instability (MSI) versus
 3134

3132 microsatellite stability (MSS) in gastrointestinal cancer, supporting research in histopathology image
 3133 analysis with CNNs and transfer learning.
 3134

3135 **6. breast_cancer_segmentation.** The *Breast Cancer Cell Segmentation* dataset (MVD, 2024b)
 3136 contains 58 H&E stained histopathology images with expert annotations for breast cancer cell
 3137 detection and segmentation. The challenging task is cell segmentation for subsequent classification
 3138 into benign and malignant cells, supported by ground truth data for algorithm development.
 3139

3140 **7. ovarian_cancer.** The *Ovarian Cancer & Subtypes Dataset Histopathology* (Pieces,
 3141 2024) contains histopathology images representing four subtypes of ovarian cancer as well
 3142 as non-cancerous tissue. The dataset is referenced as: Kasture, Kokila (2021), “Ovarian-
 3143 Cancer&SubtypesDatasetHistopathology”, Mendeley Data, V1, doi: 10.17632/kztymsrjx9.1.
 3144

3145 **8. breast_cancer_hist.** The *Breast Cancer Histopathology* dataset (Kumar, 2024) includes JPG
 3146 images labeled as benign or malignant, supporting automated breast cancer classification from
 3147 histopathological images.
 3148

3149 **9. BreCaHAD.** The *BreCaHAD* (Breast Cancer Histopathological Annotation and Diagnosis)
 3150 dataset (TruthIsNeverLinear, 2024) comprises 162 annotated H&E-stained images, supporting au-
 3151 tomated classification of histological structures into six classes: mitosis, apoptosis, tumor nuclei,
 3152 non-tumor nuclei, tubule, and non-tubule. See: <https://bmcresnotes.biomedcentral.com/articles/10.1186/s13104-019-4121-7>.
 3153

3154 **10. melanoma.** The *melanoma* dataset (Haashaatif, 2024) is designed for the development of deep
 3155 learning models for nuclei and tissue segmentation in melanoma H&E-stained histopathology. It
 3156 addresses challenges of melanocyte mimicry and includes nuclei and tissue annotations to facilitate
 3157 studies on tumor-infiltrating lymphocytes and predictive/prognostic tasks.
 3158

3159 **11. choledoch.** The *Choledoch* dataset (HFUTYBX, 2024) introduces both microscopy hyper-
 3160 spectral and color images for cholangiocarcinoma, including 880 scenes from 174 individuals (689
 3161 partial cancer, 49 full cancer, 142 non-cancer). All cancer areas are precisely labeled by ex-
 3162 perienced pathologists. More information is available in: <https://ieeexplore.ieee.org/document/8869757>. The dataset includes suggested train/val/test splits.
 3163

3164 **12. histopath-sn.** The *histopath-sn* Kaggle dataset (Feng, 2024) focuses on classifying patches and
 3165 patients from bronchus and lung samples. Both images and labels are provided, with recommended
 3166 train and test splits given in `train_labels.csv` and `test_labels.csv`.
 3167

3168 **13. ULMS.** The *Uterine Leiomyosarcoma (ULMS)* dataset (Lee, 2024) targets mitosis detection in
 3169 ULMS, the most common uterine sarcoma. Images were collected in collaboration with pathologists
 3170 and annotated for mitosis, aiding AI-based approaches for automatic mitosis detection and grading.
 3171

3172 **14. MonuSeg.** The *MonuSeg* dataset (Dinh, 2024) comprises 24 training images (originally 30,
 3173 1000 × 1000 pixels) with 21,623 annotated nuclei from seven organs, and a test set of 58 images (8
 3174 from MonuSeg, 50 from the TNBC dataset). Annotations were made by one expert pathologist and
 3175 two research fellows using consensus peer review.
 3176

3177 **15. kmc_kidney.** The *KMC Kidney Histopathology* dataset (Dwivedi, 2024) includes non-cancerous
 3178 (Grade-0) and cancerous (Grades 1-4) images of renal clear cell carcinoma, collected at Kasturba
 3179 Medical College (KMC), India. Images were stained with H&E and labeled according to grade,
 3180 supporting studies in kidney cancer histopathology.
 3181

3182 **16. histo-img-text.** The *histo-img-text* dataset (Reasat, 2024) comprises histopathology image-text
 3183 pairs, including over 32k PNGs, 40k JPGs, and a CSV file with 217,052 captioned image entries. The
 3184 dataset is designed for multimodal studies, such as image-to-text and vision-language modeling.
 3185

3186 **17. cellnet.** *CellNet* is a large, curated dataset (Capocyan, 2024) featuring over 120,000 high-
 3187 quality medical images from more than 20 organ/cancer classes. Images were aggregated from
 3188 diverse repositories and medical labs, supporting comprehensive research in computational pathology.
 3189

3190 **18. PanNuke.** The *PanNuke* dataset (Lad, 2024) is a semi-automatically generated nuclei instance
 3191 segmentation and classification dataset. It covers 481 visual fields across 19 tissue types, containing
 3192 205,343 labeled nuclei with segmentation masks, enabling tissue type segmentation and generalization
 3193 to new tissue domains.

3194 **19. NPC-88k-Public.** The *NPC-88k-Public* dataset (Munirah, 2024) includes 88,000 histopathology
 3195 patches from 17 whole slide images across three institutions. Annotated regions include normal,
 3196 lymphoid hyperplasia (LHP), nasopharyngeal inflammation (NPI), and nasopharyngeal carcinoma
 3197 (NPC), with concordance among at least two pathologists.

3198 **20. EBHI.** The *EBHI* dataset (Alibabaei78, 2024) comprises 4,456 histopathology images and
 3199 corresponding ground truth segmentations, including normal, polyp, low-grade and high-grade
 3200 intraepithelial neoplasia, serrated adenoma, and adenocarcinoma. Images are paired with ground
 3201 truth labels to support segmentation and classification research.

3202 **MRI:**

3203 Our collection of 28 Magnetic Resonance Imaging (MRI) datasets supports a diverse array of machine
 3204 learning tasks such as binary and multi-class classification, anatomical and pathological segmentation,
 3205 anomaly detection, multi-modal image registration, and physiological parameter estimation. The
 3206 included datasets range from unlabeled brain scans (*Brain MRI Images*) to richly annotated clinical
 3207 benchmarks such as *Brats*, *WMH*, and *ISLES 2015*, covering tumor segmentation, white matter lesion
 3208 detection, and ischemic stroke assessment. Cardiac datasets like *ACDC* facilitate diagnosis of specific
 3209 heart conditions, while spine-related datasets such as the *RSNA 2024 Lumbar Spine Challenge* and
 3210 *Foraminal Stenosis MRI* target degenerative spinal diseases. Other specialized collections, including
 3211 *Facial MRI*, *Prostate MRI*, and multi-modal datasets (e.g., *MRI-PET Brain Scans*), enable cross-
 3212 domain generalization and analysis. Together, this curated set of MRI datasets provides a foundation
 3213 for training and benchmarking AI systems across a broad range of anatomical regions and diagnostic
 3214 challenges.

3215 **1. Brain MRI Images** A Kaggle dataset (bra, b) containing diverse brain MRI images sourced
 3216 from multiple datasets, offering a range of anatomical variations and imaging contrasts.

3217 **2. Alzheimer Classification** Brain MRI dataset (alz) labeled for Alzheimer's disease classification
 3218 into four categories: Mild Demented, Moderate Demented, Non-Demented, and Very Mild Demented.

3219 **3. Brain Cancer** Brain MRI images (bra, a) collected from hospitals in Bangladesh for classification
 3220 into Brain Glioma, Brain Meningioma, and Pituitary Tumor classes.

3221 **4. Brain Tumour** A labeled brain tumor dataset (bra, e) for binary classification (tumor vs.
 3222 non-tumor) and unlabeled prediction samples for testing.

3223 **5. 4 Class Brain Tumour** A brain MRI dataset (bra, d) for classifying tumors into Benign,
 3224 Malignant, and Pituitary types.

3225 **6. Heat MRI Left Atrial Segmentation** A segmentation dataset (hea) of left atrial structures in
 3226 cardiac MRI provided by King's College London.

3227 **7. PMRAM** MRI brain cancer dataset (pmr) with four classes (Glioma, Meningioma, Pituitary, No
 3228 Tumor), standardized to 512×512 resolution and augmented from 1600 base images.

3229 **8. Hippocampal Sparing** Unlabeled DICOM-format MRI slices (hip) of 25 patients for hippocam-
 3230 pal sparing studies, organized per patient.

3240 **9. Spine** Spine MRI scans (spi) from a single patient with labeled dystrophic anomalies and
 3241 accompanying radiology reports.
 3242

3243 **10. Brain Tumour CT MRI** A brain tumor dataset (bra, c) composed of both MRI and CT images,
 3244 labeled for tumor detection and suitable for training diagnostic models.
 3245

3246 **11. BraTS 2019** Multimodal brain MRI dataset (Menze et al., 2014) (T1, T1Gd, T2, FLAIR) with
 3247 expert segmentations for tumor subregions, formatted as NIfTI (.nii.gz) files.
 3248

3249 **12. Bone Fractures MRI X-ray** Multi-modal dataset (hbf) including MRI and X-ray scans for
 3250 bone fracture detection across different body regions.
 3251

3252 **13. Alzheimer Detection** Preprocessed MRI scans (LaMontagne et al., 2019) from the OASIS-1
 3253 dataset labeled for Alzheimer's detection tasks.
 3254

3255 **14. Stroke Head MRI** MRI brain scans (str) with segmentations of stroke lesions from patients
 3256 with cerebrovascular conditions.
 3257

3258 **15. MRI PET Brain Scans** Paired MRI and PET DICOM scans (mri) for brain tumors, aimed at
 3259 multi-modal registration and Dice score evaluation.
 3260

3261 **16. OASIS-1** Processed MRI scans of 1688 subjects across Alzheimer's Disease (AD), Cognitively
 3262 Normal (CN), and Mild Cognitive Impairment (MCI) groups (oas).
 3263

3264 **17. Abdomen MRI** Abdominal MRI dataset (abd) with object detection annotations and bounding
 3265 boxes in CSV format.
 3266

3267 **18. Facial MRI** Facial MRI scans (fac) including sagittal and axial slices for anomaly detection,
 3268 segmentation, and 3D anatomical modeling.
 3269

3270 **19. Prostate** Multi-parametric prostate MRI scans (pro, a) with manual segmentations for clinical
 3271 segmentation research.
 3272

3273 **20. Glioma** TCGA-LGG-based MRI dataset (gli) for low-grade glioma detection with segmentation
 3274 masks and associated genomics metadata.
 3275

3276 **21. Phantom** Longitudinal MRI dataset (pha) of a single healthy subject scanned on 116 scanners
 3277 over 2.5 years to analyze scanner variability.
 3278

3279 **22. ACDC: Automated Cardiac Diagnosis Challenge Dataset** The ACDC (Bernard et al.,
 3280 2018) dataset consists of cine-MRI scans, categorized into five balanced cardiac pathology classes:
 3281 Normal (NOR), Myocardial Infarction (MINF), Dilated Cardiomyopathy (DCM), Hypertrophic
 3282 Cardiomyopathy (HCM), and Abnormal Right Ventricle (ARV). Each class is defined by specific
 3283 clinical parameters such as ejection fraction, wall thickness, and ventricular volumes, supporting
 3284 robust machine learning development for automated cardiac function assessment.
 3285

3286 **23. Foraminal Stenosis MRI Dataset** This dataset (for) comprises high-resolution lumbar spine
 3287 MRI scans with segmentation masks and foraminal measurements, aimed at detecting and analyzing
 3288 foraminal stenosis. It supports tasks such as nerve channel size analysis, stenosis classification, and
 3289 monitoring of spinal degenerative conditions, enabling precise anatomical assessment and aiding in
 3290 early diagnosis and treatment planning.
 3291

3292 **24. RSNA 2024 Lumbar Spine Degenerative Classification Challenge** This RSNA-ASNR (RSN)
 3293 dataset includes five lumbar spine degenerative conditions—Left/Right Neural Foraminal Narrowing,
 3294 Left/Right Subarticular Stenosis, and Spinal Canal Stenosis—using lumbar spine MRI. The dataset
 3295 includes severity scores (Normal/Mild, Moderate, Severe) across five disc levels (L1/L2 to L5/S1),
 3296 enabling automated classification to support diagnosis and treatment planning for lower back pain
 3297 and related conditions.
 3298

3294 **25. ATLAS v2.0** The Anatomical Tracings of Lesions After Stroke (ATLAS) v2.0 (Liew et al.,
 3295 2022) dataset provides manually segmented T1-weighted MRI scans of individuals with stroke lesions.
 3296 It includes lesion masks and anatomical metadata for over 600 subjects, with the aim of facilitating
 3297 the development and benchmarking of automated stroke lesion segmentation methods.
 3298

3299 **26. BraTS** The Brain Tumor Segmentation (BraTS) dataset provided through the Medical Segmen-
 3300 tation Decathlon (MSD), comprises multi-modal MRI scans (T1, T1-Gd, T2, and FLAIR) of glioma
 3301 patients with expert annotations of tumor sub-regions including the enhancing tumor, peritumoral
 3302 edema, and necrotic core.
 3303

3304 **27. WMH** The White Matter Hyperintensities (WMH) dataset (wmh) consists of T1 and FLAIR
 3305 MRI scans from multiple institutions with voxel-wise annotations of WMH regions. Originally
 3306 compiled for the WMH Segmentation Challenge at MICCAI 2017, the dataset captures variability
 3307 across scanners and populations, making it a robust benchmark for automated WMH detection
 3308 methods.
 3309

3310 **28. ISLES 2015 (SISS)** The Ischemic Stroke Lesion Segmentation (ISLES) 2015 challenge dataset
 3311 (isl), specifically the Sub-Acute Ischemic Stroke Lesion Segmentation (SISS) subtask, offers multi-
 3312 modal MRI scans (including FLAIR, T1, DWI) with corresponding lesion masks for patients in the
 3313 subacute phase post-stroke. It supports the development of methods for accurate ischemic stroke
 3314 lesion segmentation and includes cases with diverse lesion locations and volumes.
 3315

FUNDUS:

3317 Our Fundus image datasets span a broad range of tasks and clinical applications, reflecting the
 3318 diagnostic richness of retinal imaging. These include segmentation datasets such as *Drishti-GS*,
 3319 *RIMONE*, and *ONH Segmentation* for optic disc/cup analysis in glaucoma, and vessel segmentation
 3320 benchmarks like *DRIVE* and *CHASE_DB1* for vascular assessment. Classification datasets such as
 3321 *APROS*, *MESSIDOR*, and *ARIA* support diabetic retinopathy grading, while multi-label datasets like
 3322 *RFMID* and *ODIR-5K* address a broader set of ocular diseases. Lesion-level annotations in datasets
 3323 like *IDRiD* and *E-Ophtha* enable fine-grained detection of diabetic pathologies. Additionally, niche
 3324 datasets such as *e-ROP*, *Ocular Toxoplasmosis*, and *AMDP* target rare or longitudinal conditions.
 3325 Others focus on preprocessed imaging (*CLAHE + ESRGAN Split FD*) or multi-modal metadata
 3326 (*SMDG*, *DrHagis*). This diversity supports robust benchmarking across segmentation, classification,
 3327 enhancement, and multimodal learning, forming the backbone of data-driven ophthalmic model
 3328 development.
 3329

3329 **1. Drishti-GS** This dataset (Sivaswamy et al., 2014) is used for glaucoma detection, providing
 3330 optic disc and cup segmentation masks. It supports both segmentation and glaucoma classification
 3331 tasks.
 3332

3333 **2. STARE** The STARE dataset (STA) is used for retinal disease diagnosis and retinal vessel
 3334 segmentation. Its main tasks include vessel segmentation and lesion detection.
 3335

3336 **3. IDRiD** The Indian Diabetic Retinopathy Image Dataset (IDRiD) (ind) provides pixel-level
 3337 annotations for diabetic retinopathy (DR) lesions. It is used for lesion segmentation and DR grading.
 3338

3339 **4. DR** This dataset (DR) is used for classifying diabetic retinopathy across 5 severity levels.
 3340

3341 **5. RIMONE** A glaucoma dataset (Fumero et al., 2011) providing optic disc and cup annotations,
 3342 mainly used for segmentation and glaucoma classification.
 3343

3344 **6. REFUGE** A unified glaucoma evaluation dataset (ref), widely used for optical disc/cup segmen-
 3345 tation and glaucoma classification.
 3346

3347 **7. CHASE_DB1** This dataset (cha) contains child retinal images with annotated vessels. It is
 3348 primarily used for vessel segmentation tasks.
 3349

3348 **8. E-Ophtha** Designed for diabetic retinopathy research, this dataset (Decenciere et al., 2013)
 3349 includes images annotated for exudates and hemorrhages, supporting lesion detection.
 3350

3351 **9. ARIA** A retinal image dataset used in diabetic retinopathy screening. It is mainly employed for
 3352 DR classification.
 3353

3354 **10. IOSTAR** A dataset of multi-modal retinal images, particularly used for optic disc segmentation
 3355 tasks.
 3356

3357 **11. HRF** The High-Resolution Fundus dataset is used for both vessel and optic disc segmentation,
 3358 offering detailed structural annotations.
 3359

3360 **12. LES-AV** This dataset supports artery and vein classification, distinguishing vessel types in
 3361 fundus images.
 3362

3363 **13. PRIME-FP20** It is a high-resolution dataset of fundus images used for optic disc segmentation.
 3364

3365 **14. RIGA+** This is a glaucoma dataset derived from multiple sources, used for optic disc and cup
 3366 segmentation.
 3367

3368 **15. APTOS** It is part of the Kaggle Diabetic Retinopathy Challenge (2019), this dataset is used to
 3369 grade DR severity from fundus images.
 3370

3371 **16. MESSIDOR** It is a classic and widely used diabetic retinopathy dataset, primarily for classifi-
 3372 cation tasks.
 3373

3374 **17. DRIVE** It is one of the earliest vessel segmentation datasets, often used as a benchmark in
 3375 fundus segmentation.
 3376

3377 **18. ORIGA** The ORIGA dataset provides optic disc and cup annotations for segmentation task and
 3378 glaucoma detection.
 3379

3380 **19. ODIR-5K** The ODIR (Ocular Disease Intelligent Recognition) dataset contains over 5,000
 3381 retinal fundus images with multi-label annotations for eight ocular diseases, including diabetic
 3382 retinopathy, glaucoma, cataract, AMD, hypertension, and others. It supports multi-label classification
 3383 tasks.
 3384

3385 **20. RFMID** The Retinal Fundus Multi-Disease Image Dataset (RFMID) includes 3,200+ images
 3386 annotated for 19 different conditions. It is intended for multi-label classification tasks and supports
 3387 the development of fundus-based diagnostic models for diverse ocular diseases.
 3388

3389 **21. MESSIDOR-2 DF** MESSIDOR-2 is the second edition of the MESSIDOR diabetic retinopathy
 3390 dataset. It includes fundus images with diabetic retinopathy severity labels.
 3391

3392 **22. Glaucoma datasets (EYE-PACS)** EYE-PACS is a large-scale dataset used primarily for
 3393 diabetic retinopathy grading in the Kaggle challenge.
 3394

3395 **23. Retina blood vessel segmentation dataset** This fundus dataset is used for vessel segmentation.
 3396

3397 **24. DR Diagnosis Dataset** This dataset is used for classifying diabetic retinopathy severity based
 3398 on retinal fundus images.
 3399

3400 **25. DDR Dataset** The Diabetic Retinopathy Detection from Retina Images (DDR) dataset includes
 3401 fundus images annotated for DR severity and pixel-level lesion types (e.g., exudates, hemorrhages).
 It supports both classification and lesion segmentation tasks.

3402 **26. Hypertensive Retinopathy** This dataset contains fundus images annotated for signs of hypertensive retinopathy. While rare and usually hospital-specific, it is used for classification and grading of HR severity.

3403
3404
3405
3406 **27. SUSTECH + SYSU Dataset** This entry combines data from SUSTech and Sun Yat-sen University (SYSU), curated for research in glaucoma, diabetic retinopathy, and related diseases. It supports classification tasks across multiple disease categories.

3407
3408
3409
3410 **28. RITE** The Retinal Images vessel Tree Extraction (RITE) dataset, derived from DRIVE, includes ground truth for artery and vein segmentation. It is used to differentiate between arterial and venous vessels in retinal images.

3411
3412
3413
3414 **29. CLAHE + ESRGAN Split FD** This dataset represents a preprocessed variant of fundus images where contrast enhancement (CLAHE) and super-resolution techniques (ESRGAN) have been applied. It is used to improve image quality for downstream classification tasks.

3415
3416
3417 **30. Myopia Image Dataset** This dataset consists of retinal fundus images labeled for myopia classification.

3418
3419
3420 **31. ACRIMA** ACRIMA is fundus dataset used for glaucoma detection.

3421
3422
3423 **32. and 33. Retina Fundus Dataset (CHASE_DB1, DRIVE)** CHASE_DB1 and DRIVE are fundus datasets used for retinal vessel segmentation, i.e., for segmenting blood vessels in fundus images.

3424
3425
3426 **34. Cataract Classification Dataset** This is used for binary classification of cataract presence in fundus images.

3427
3428
3429 **35. MURED** The Multicenter Retinal Disease Dataset (MURED) aggregates retinal images across multiple institutions and includes annotations for diabetic retinopathy, glaucoma, age-related macular degeneration (AMD), and other conditions. It is primarily used for multi-class classification of retinal diseases.

3430
3431
3432
3433
3434 **36. Optic Disc Cup Fundus Image** This dataset contains annotations for optic disc and cup structures. These datasets are used for segmentation tasks and for calculating cup-to-disc ratio, an important indicator in glaucoma diagnosis.

3435
3436
3437
3438 **37. ROFT** This is a retinal and ocular fundus image dataset with 8 disease labels for fundus images - normal, diabetes, glaucoma, cataract, age-related macular degeneration, hypertension, pathological myopia and other diseases/abnormalities. It also has 7 labels for OCT: age-related macular degeneration, diabetic macular edema, epiretinal membrane, normal, retinal artery occlusion, retinal vein occlusion, vitreomacular interface diseases.

3439
3440
3441
3442
3443 **38. Eye Disease Image Dataset** A fundus dataset for detection of eye-related 10 conditions - central serous chorioretinopathy, diabetic retinopathy, disc edema, glaucoma, healthy, macular scar, myopia, pterygium, retinal detachment, and retinitis pigmentosa.

3444
3445
3446
3447 **39. FIVES** The FIVES dataset (Fundus Image Vessel Extraction and Segmentation) is used for vessel segmentation tasks. It provides pixel-level annotations for blood vessel structures.

3448
3449
3450
3451 **40. AMDP Dataset** This refers to the Age-related Macular Degeneration Prediction dataset which is longitudinal ophthalmic dataset.

3452
3453
3454
3455 **41. AGAR 300** A Microaneurysms Fundus Dataset that consists of color fundus images showing signs of microaneurysms for early DR detection.

3456 **42. SMDG** It is a standardized fundus glaucoma dataset consisting of full-fundus glaucoma images
 3457 with image metadata on optic disc/cup segmentation and blood vessel segmentation.
 3458

3459 **44. Fundus segmentation dataset** It is a unified retinal image dataset for assessing glaucoma with
 3460 reference segmentation labels of optic disc and cup.
 3461

3462 **45. Hypertensive retinopathy dataset** It is a fundus dataset for binary classification regarding
 3463 presence or absence of hypertensive retinopathy.
 3464

3465 **46. DR grading dataset** It is a fundus dataset for grading the severity of diabetic retinopathy.
 3466

3467 **47. G1020 dataset** It is a fundus dataset for glaucoma classification and contain 1020 high
 3468 resolution colour fundus images. It also provides annotations for glaucoma diagnosis, optic disc and
 3469 cup segmentation, vertical cup to disc ratio, etc.
 3470

3471 **48. Ocular Toxoplasmosis dataset** It is a fundus dataset used for detection of Toxoplasmosis
 3472 chorioretinitis and has three classes - healthy eye, active and inactive chorioretinitis.
 3473

3474 **49. ARIA dataset** It is a fundus dataset used for detection of any of three classes: healthy, AMD
 3475 and Diabetes.
 3476

3477 **50. Fundus 4 categories dataset** It is a fundus dataset used for detection of normal, cataract,
 3478 glaucoma and diabetic retinopathy.
 3479

3480 **51. ONH Segmentation dataset** It is an optic disc and cup mask segmentation fundus dataset
 3481

3482 **52. DrHagis dataset** It is a fundus dataset for detection of diabetic retinopathy, hypertension,
 3483 age-related macular degeneration and glaucoma.
 3484

3485 **53. Driona DB dataset** It is a fundus dataset for optic disc segmentation.
 3486

3487 **54. Cattle Retinal Fundus Images** A unique dataset featuring retinal fundus images from cattle,
 3488 useful for comparative studies and veterinary ophthalmology research.
 3489

3490 **55. Preprocessed Eye Diseases Fundus Images** It offers preprocessed fundus images enhanced
 3491 using techniques like CLAHE and ESRGAN, facilitating improved classification performance.
 3492

3493 **56. Retina Fundus Image Registration Dataset (FIRE)** It comprises 129 retinal images forming
 3494 134 image pairs, designed for evaluating image registration algorithms.
 3495

3496 **57. 1000 Fundus Images with 39 Categories** This dataset comprises 1,000 fundus images
 3497 categorized into 39 distinct classes, offering a diverse set for multi-class classification tasks.
 3498

3499 **58. PAPILA: Retinal Fundus Images Dataset** The PAPILA dataset includes fundus images and
 3500 clinical data from both eyes of individual patients for glaucoma assessment. It provides optic disc
 3501 and cup segmentations, along with patient-level glaucoma labels derived from clinical evaluations.
 3502

3503 **59. Diabetic Retinopathy Diagnosis Dataset** A large-scale retinal image dataset designed for the
 3504 diagnosis of diabetic retinopathy, supporting medical image analysis and automated disease grading.
 3505

3506 **60. Vessel Tree Extraction Dataset** This dataset supports comparative research on artery and vein
 3507 segmentation or classification in retinal fundus images, facilitating the development and benchmarking
 3508 of vessel-type analysis methods.
 3509

3510 **61. DiaRetDB1: Diabetic Retinopathy Benchmark Dataset** DiaRetDB1 includes retinal fundus
 3511 images with expert-annotated ground truth for key lesions such as hard and soft exudates,
 3512 microaneurysms, and hemorrhages, along with both the original images and raw annotation data.
 3513

3510
 3511 **62. SynFundus** The SynFundus is a synthetic fundus dataset includes annotations for eleven
 3512 ocular diseases: diabetic retinopathy, age-related macular degeneration, anomalies of the optic
 3513 nerve, choroidal retinal pathology, degenerative and pathological myopia, diabetic macular edema,
 3514 epimacular membrane, glaucoma, hypertensive retinopathy, and retinal vein occlusion. These
 3515 conditions cover a broad range of structural and vascular retinal abnormalities, supporting diverse
 3516 diagnostic research in ophthalmology.

3517
 3518 **63. AIROGS** The AIROGS dataset (De Vente et al., 2023) comprises fundus photographs from
 3519 diverse ethnicities and imaging devices. It supports two main tasks: referable glaucoma classification
 3520 and detection of ungradable images to simulate real-world screening conditions.

3521 C.2 SAMPLE DATASET DESCRIPTION FILES:

3522 Sample dataset description files are shown in Listings 6-8. The datasets are then partitioned into
 3523 different clients and utilized by the client selector agents to decide whether to choose the client for
 3524 federated analysis.

3525 Listing 6: Dataset Descriptions for Dermatology Modality

```

3526
3527
3528
3529 [
3530   {
3531     "Dataset Name": "augmented_skin_condition_dataset_kaggle",
3532     "Dataset Description": "augmented_skin_condition_dataset_kaggle
3533       is a skin disease classification dataset containing images of
3534       six different dermatological conditions: 'Acne', 'Carcinoma
3535       ', 'Eczema', 'Keratosis', 'Milia', and 'Rosacea'. It contains
3536       six subfolders, with each subfolder containing images of the
3537       corresponding class (disease) specified in the name of the
3538       subfolder. ",
3539     "Dataset
3540       Path": "skin_dataset/augmented_skin_condition_dataset_kaggle"
3541   },
3542   {
3543     "Dataset Name": "DDI_skin_dataset",
3544     "Dataset Description": "DDI_skin_dataset is a skin cancer
3545       classification dataset with diverse skin tone representation
3546       that contains 1 subfolder 'images' and 2 CSV files. Focus on
3547       the columns: 'DDI_file' (for the image path) and 'malignant'
3548       (the class label) of the csv file 'ddi_metadata.csv'. 'True'
3549       in 'malignant' column means malignant whereas 'False' means
3550       benign. ",
3551     "Dataset Path": "skin_dataset/DDI_skin_dataset"
3552   },
3553   {
3554     "Dataset Name": "Derma7PT",
3555     "Dataset Description": "Derma7PT is a skin disease classification
3556       dataset containing a subfolder 'images' and a csv file 'meta
3557       .csv'. Focus on the columns 'clinic' and 'derm' for the image
3558       file path as well as the column 'diagnosis' of the csv file
3559       that has 10 disease types: 'basal cell carcinoma', 'nevus', 'dermatofibroma', 'lentigo', 'melanoma', 'melanoma metastasis', 'melanosis', 'miscellaneous', 'seborrheic keratosis', 'vascular lesion'. ",
3560     "Dataset Path": "skin_dataset/Derma7P"
3561   },
3562   {
3563     "Dataset Name": "Dermatology_tabular dataset",
3564     "Dataset Description": "Dermatology_tabular dataset is a tabular
3565       (non-image) dataset containing clinical features for
3566       diagnosing skin diseases. ",
3567     "Dataset Path": "skin_dataset/Dermatology_tabular dataset"
3568   },
3569   {

```

```

3564     "Dataset Name": "Dermis",
3565     "Dataset Description": "Dermis is a skin disease dataset with
3566         benign and malignant cases, supporting both classification
3567         and segmentation tasks. It has two sub-folders 'benign' and 'melanoma'. In each of these sub-folders, we have two sub-
3568         folders 'contour' (that has the segmentation masks) and 'images' (that has the original images). ",
3569     "Dataset Path": "skin_dataset/Dermis"
3570 },
3571 },
3572 {
3573     "Dataset Name": "Dermnet",
3574     "Dataset Description": "Dermnet contains a very broad collection
3575         of skin disease images. It has 23 sub-folders covering 23
3576         disease categories namely 'Acne and Rosacea', 'Actinic
3577         Keratosis Basal Cell Carcinoma and other Malignant Lesions',
3578         'Atopic Dermatitis Photos', 'Bullous Disease Photos', 'Cellulitis Impetigo and other Bacterial Infections', 'Eczema
3579         Photos', 'Exanthems and Drug Eruptions', 'Hair Loss (Alopecia
3580         ) and other Hair Diseases', 'Herpes HPV and other STDs Photos
3581         ', 'Light Diseases and Disorders of Pigmentation', 'Lupus and
3582         other Connective Tissue Diseases', 'Melanoma Skin Cancer
3583         Nevi and Moles', 'Nail Fungus and other Nail Disease', 'Poison Ivy Photos and other Contact Dermatitis', 'Psoriasis
3584         pictures and Lichen Planus and related Diseases', 'Scabies
3585         Lyme Disease and other Infestations and Bites', 'Seborrheic
3586         Keratoses and other Benign Tumors', 'Systemic Disease',
3587         'Tinea Ringworm Candidiasis and other Fungal Infections',
3588         'Urticaria Hives', 'Vascular Tumors', 'Vasculitis Photos', 'Warts Molluscum and other Viral Infections'.
3589     "Dataset Path": "skin_dataset/Dermnet"
3590 },
3591 {
3592     "Dataset Name": "Dermquest",
3593     "Dataset Description": "Dermquest is a skin disease
3594         classification and segmentation dataset containing images of
3595         benign and malignant skin diseases. It has two sub-folders 'benign' and 'melanoma'. In each of these sub-folders, we have
3596         two sub-folders 'contour' (that has the segmentation masks)
3597         and 'images' (that has the original images). "
3598     "Dataset Path": "skin_dataset/Dermquest"
3599 },
3600 {
3601     "Dataset Name": "fitzpatrick17k",
3602     "Dataset Description": "fitzpatrick17k is a large skin lesion
3603         dataset with a wide range of dermatological diseases. It has
3604         a sub-folder 'finalfitz17k' which contains all images and two
3605         csv files 'fitzpatrick17k_disease.csv' and 'Fitzpatrick17k_morphology.csv'. Focus on the column 'md5hash'
3606         for filename and the column 'three_partition_label' that
3607         contains three disease labels: 'non-neoplastic', 'benign', 'malignant' in the file 'fitzpatrick17k_disease.csv'.
3608     "Dataset Path": "skin_dataset/fitzpatrick17k"
3609 },
3610 {
3611     "Dataset Name": "ISIC2018_HAM10000",
3612     "Dataset Description": "ISIC2018_HAM10000 is a skin lesion
3613         classification and segmentation dataset. It has a sub-folder
3614         'HAM10000_images_combined_600x450' that contains original
3615         images as well as a sub-folder 'HAM10000_segmentations_mask'
3616         that contains the corresponding segmentation masks. The
3617         classification labels can be found in the 'dx' column of the
3618         csv file 'ISIC2018_Task3_Test_GroundTruth.csv' including
3619         'Melanocytic Nevus (nv)', 'Benign Keratosis-like Lesions (bkl)'
3620         ', 'Melanoma (mel)', 'Basal Cell Carcinoma (bcc)', 'Actinic
3621         Keratosis / Bowen's Disease (akiec)', 'Vascular Lesions (vasc'
3622

```

```

3618     ), 'Dermatofibroma (df)'. The corresponding image names can
3619     be found in the column 'image_id' of the same csv file. ",
3620     "Dataset Path": "skin_dataset/ISIC2018_HAM10000"
3621   },
3622   {
3623     "Dataset Name": "ISIC_2016",
3624     "Dataset Description": "ISIC_2016 is a skin lesion dataset for
3625       classification and segmentation, focused on skin cancer
3626       detection. It has two sub-folders 'ISBI2016_ISIC_images' that
3627       contain original images and 'ISBI2016_ISIC_segmentation_mask'
3628       that has segmentation masks. The csv file 'ISBI2016_ISIC_binary_classification_Training_GroundTruth.csv'
3629       has two columns - the first column being image names and
3630       second column being binary disease labels: 'benign' and '
3631       malignant'. ",
3632     "Dataset Path": "skin_dataset/ISIC_2016"
3633   },
3634   {
3635     "Dataset Name": "ISIC_2017",
3636     "Dataset Description": "ISIC_2017 is a skin lesion classification
3637       and segmentation dataset with a focus on melanoma and
3638       seborrheic keratosis diagnosis. It has two sub-folders: 'images'
3639       that contain original images and 'Segmentation_masks'
3640       that has segmentation masks. There is a csv file 'ISIC-2017
3641       _GroundTruth' with the columns 'image_id' that contains image
3642       filenames, 'melanoma' that contains binary labels
3643       corresponding to presence (1) and absence (0) of melanoma,
3644       and 'seborrheic keratosis' that contains binary labels
3645       corresponding to presence (1) and absence (0) of seborrheic
3646       keratosis. ",
3647     "Dataset Path": "skin_dataset/ISIC_2017"
3648   },
3649   {
3650     "Dataset Name": "ISIC_2019",
3651     "Dataset Description": "ISIC_2019 is an extended skin disease
3652       classification dataset. It has one sub-folder: 'images' that
3653       contain original images. In the CSV file 'ISIC_2019_Training_GroundTruth.csv', the 'image' column
3654       contains the image file names and 9 other columns represent
3655       the presence (1) or absence (0) of 9 classes namely Melanoma
3656       (MEL), Nevus (NV), Basal Cell Carcinoma (BCC), Actinic
3657       Keratosis / Bowen's Disease (AK), Benign Keratosis-like
3658       Lesions (BKL), Dermatofibroma (DF), Vascular Lesions (VASC),
3659       Squamous Cell Carcinoma (SCC) and Unknown (UNK). ",
3660     "Dataset Path": "skin_dataset/ISIC_2019"
3661   },
3662   {
3663     "Dataset Name": "ISIC_2020",
3664     "Dataset Description": "ISIC_2020 is a binary classification
3665       dataset of skin lesions (benign vs malignant). It has one sub
3666       -folder: 'images' that contain original images. In the CSV
3667       file 'ISIC_2020_Training_GroundTruth.csv', the 'image_name'
3668       column contains the image file names and the '
3669       benign_malignant' column contains the corresponding disease
3670       labels on malignant or benign. ",
3671     "Dataset Path": "skin_dataset/ISIC_2020"
3672   },
3673   {
3674     "Dataset Name": "ISIC_2024",
3675     "Dataset Description": "ISIC_2024 is an updated ISIC skin disease
3676       dataset primarily for melanoma classification (binary:
3677       benign vs malignant). It has one sub-folder: 'images' that
3678       contain original images. In the CSV file '
3679       ISIC_2024_Training_GroundTruth.csv', the 'isic_id' column
3680       contains the image file names and the 'malignant' column

```

```

3672     contains the corresponding disease labels on malignant or
3673     benign. '0' means benign and '1' means malignant. ",
3674     "Dataset Path": "skin_dataset/ISIC_2024"
3675   },
3676   {
3677     "Dataset Name": "Mednode",
3678     "Dataset Description": "Mednode is a skin disease dataset for
3679       binary classification. It has 2 sub-folders covering 2
3680       disease categories namely melanoma and nevus. ",
3681     "Dataset Path": "skin_dataset/Mednode"
3682   },
3683   {
3684     "Dataset Name": "Monkeypox_Skin_Image_Dataset",
3685     "Dataset Description": "Monkeypox_Skin_Image_Dataset is a dataset
3686       for skin disease classification and has four sub-folders (
3687       with data belonging to the corresponding disease category)
3688       named: 'Chickenpox', 'Measles', 'Monkeypox', and 'Normal'. ",
3689     "Dataset Path": "skin_dataset/Monkeypox_Skin_Image_Dataset"
3690   },
3691   {
3692     "Dataset Name": "PAD_UFES_20",
3693     "Dataset
3694       Description": "PAD_UFES_20 is a skin disease classification
3695       dataset. It contains a sub-folder 'images' containing the
3696       original images and a csv file called 'metadata.csv' that
3697       contains the image ids under the column 'img_id' and disease
3698       labels under the column 'diagnostic' which contains 6 disease
3699       categories with corresponding abbreviations: Melanoma (MEL),
3700       Melanocytic Nevus (NEV), Basal Cell Carcinoma (BCC), Actinic
3701       Keratosis / Bowen's Disease (ACK), Seborrheic Keratosis (SEK
3702       ), and Squamous Cell Carcinoma (SCC). ",
3703     "Dataset Path": "skin_dataset/PAD_UFES_20"
3704   },
3705   {
3706     "Dataset Name": "PH2Dataset",
3707     "Dataset Description": "PH2Dataset is a skin lesion
3708       classification and segmentation dataset. It has a sub-folder
3709       'PH2 Dataset images' which in turn has two sub-folders '
3710       all_dermoscopic_images' that contain all the original images
3711       and 'segmentation_mask' that contain all the segmentation
3712       masks. The folder has an xlsx file called 'PH2_dataset.xlsx'
3713       with a column called 'Image Name' that contains the image ids
3714       and a column 'Clinical Diagnosis' three disease classes :
3715       'Common Nevus', 'Atypical Nevus', and 'Melanoma' marked with '
3716       X' whenever that category is present in a given image. ",
3717     "Dataset Path": "skin_dataset/PH2Dataset"
3718   },
3719   {
3720     "Dataset Name": "scin_dataset",
3721     "Dataset Description": "scin_dataset is a multi-class skin
3722       disease classification dataset. It has a sub-folder '
3723       scin_images' that contains all the original images and two
3724       csv files. Follow the 'scin_cases.csv' file which has the
3725       image ids in the column 'case_id' and the disease classes
3726       under the 'related category' which should include the 9
3727       diseases: 'RASH', 'LOOKS_HEALTHY', 'OTHER_ISSUE_DESCRIPTION',
3728       'NAIL_PROBLEM', 'GROWTH_OR_MOLE', 'ACNE', '
3729       PIGMENTARY_PROBLEM', 'HAIR_LOSS', 'OTHER_HAIR_PROBLEM'. ",
3730     "Dataset Path": "skin_dataset/scin_dataset"
3731   },
3732   {
3733     "Dataset Name": "skin_disease_3_class",
3734     "Dataset Description": "skin_disease_3_class is a skin disease
3735       classification dataset that consists of a sub-folder 'images'
3736       which in turn has three sub-folders each consisting of one

```

```

3726     of the three classes indicated by the sub-folder name: 'acne
3727     ', 'atopic dermatitis', and 'basal cell carcinoma'. ",
3728     "Dataset Path": "skin_dataset/skin_disease_3_class"
3729   },
3730   {
3731     "Dataset Name": "skin_disease_classification_kaggle",
3732     "Dataset Description": "skin_disease_classification_kaggle is a
3733       skin disease classification dataset with a sub-folder 'files'
3734       that again contains three sub-folders each containing one of
3735       the three classes: 'acne', 'eye bags', and 'redness'. ",
3736     "Dataset Path": "skin_dataset/skin_disease_classification_kaggle"
3737   },
3738   {
3739     "Dataset Name": "skin_disease_kaggle_dataset",
3740     "Dataset
3741       Description": "skin_disease_kaggle_dataset is a skin cancer
3742       detection dataset that has 10 sub-folders for 10 disease
3743       classes with the corresponding sub-folder names: 'Atopic
3744       Dermatitis', 'Basal Cell Carcinoma (BCC)', 'Benign Keratosis-
3745       like Lesions (BKL)', 'Eczema', 'Melanocytic Nevus (NV)', '
3746       Melanoma', 'Psoriasis pictures Lichen Planus and related
3747       diseases', 'Seborrheic Keratoses and other Benign Tumors', '
3748       Tinea Ringworm Candidiasis and other Fungal Infections', and
3749       'Warts Molluscum and other Viral infections'. ",
3750     "Dataset Path": "skin_dataset/skin_disease_kaggle_dataset"
3751   },
3752   {
3753     "Dataset Name": "Skin Disease_Robo",
3754     "Dataset Description": "Skin Disease_Robo is a skin disease
3755       classification and object detection dataset. It has one sub-
3756       folder 'image' that contains all the original images and a
3757       csv file 'bounding_box_annotations.csv' with a column called
3758       'filename' that has all the image names and column 'class'
3759       that has 10 disease class labels: 'Acne', 'Atopic Dermatitis
3760       ', 'Chicken Skin', 'Eczema', 'Hansen's Disease-Leprosy', '
3761       Hansen's Disease-Leprosy-severe', 'Healthy skin', 'Psoriasis
3762       ', 'Ringworm', 'Warts'. It also contains coordinates for
3763       bounding box annotations for lesions in the columns 'xmin', '
3764       'ymin', 'xmax', and 'ymax'. ",
3765     "Dataset Path": "skin_dataset/Skin Disease_Robo"
3766   },
3767   {
3768     "Dataset Name": "skin-infection-disease-dataset",
3769     "Dataset Description": "skin-infection-disease-dataset is a skin
3770       disease classification dataset focusing on infectious skin
3771       diseases. It has 8 sub-folders consisting diseases of each
3772       category - BA-cellulitis, BA-impetigo, FU-athlete-foot, FU-
3773       nail-fungus, FU-ringworm, PA-cutaneous-larva-migrans, VI-
3774       chickenpox, VI-shingles. ",
3775     "Dataset Path": "skin_dataset/skin-infection-disease-dataset"
3776   },
3777   {
3778     "Dataset Name": "skinL2_dataset",
3779     "Dataset Description": "skinL2_dataset is a skin cancer
3780       classification dataset with 8 sub-folders containing 8
3781       classes: 'Basal-cell Carcinoma', 'Dermatofibroma', '
3782       Hemangioma', 'Melanoma', 'Nevus', 'Psoriasis', 'Seborrheic
3783       Keratosis', and 'Others'. Optional metadata is available in '
3784       PlenoISLA_DatasetV1_info.xlsx'. ",
3785     "Dataset Path": "skin_dataset/skinL2_dataset"
3786   }
3787 ]
3788

```

Listing 7: Dataset Descriptions for X-Ray Modality

```

3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
[ {
  "Dataset Name": "cov_19",
  "Dataset Description": "This is a database of chest X-ray images for COVID-19 positive cases along with Normal and Viral Pneumonia images. It has 3616 COVID-19 positive cases along with 10,192 Normal, 6012 Lung Opacity (Non-COVID lung infection), and 1345 Viral Pneumonia images and corresponding lung masks organized in different sub-folders.",
  "Dataset Path": "xray/cov_19"
},
{
  "Dataset Name": "bone_frac",
  "Dataset Description": "This dataset comprises fractured and non-fractured X-ray images covering all anatomical body regions, including lower limb, upper limb, lumbar, hips, knees, etc. The dataset is categorized into two subfolders containing fractured and non-fractured radiographic images.",
  "Dataset Path": "xray/bone_frac"
},
{
  "Dataset Name": "chest_tuberculosis_segmentation",
  "Dataset Description": "This dataset consists of 704 chest X-ray images for tuberculosis (TB) detection. The dataset contains both tuberculosis-positive and normal chest X-rays and are accompanied by lung segmentation masks (in separate subfolders) and clinical metadata as csv files.",
  "Dataset Path": "xray/chest_tuberculosis_segmentation"
},
{
  "Dataset Name": "xray/17_diseases",
  "Dataset Description": "The dataset consists of a collection of chest X-ray images in .jpg and .dcm formats. Types of diseases in the dataset: Abscess, Ards, Atelectasis, Atherosclerosis of the aorta, Cardiomegaly, Emphysema, Fracture, Hydropneumothorax, Hydrothorax, Pneumonia, Pneumosclerosis, Post inflammatory changes, Post traumatic ribs deformation, Sarcoidosis, Scoliosis, Tuberculosis and Venous congestion arranged in different subfolders.",
  "Dataset Path": "xray/17_diseases"
},
{
  "Dataset Name": "spr_age_gender",
  "Dataset Description": "SPR X-Ray Age and Gender Dataset. Used to help predict the age and gender of the patient based on the X-Ray image. Contains .png x-ray images in image subfolder with csv file containing gender and age.",
  "Dataset Path": "xray/spr_age_gender"
},
{
  "Dataset Name": "unifesp",
  "Dataset Description": "The UNIFESP X-Ray Body Part Classification Dataset. This is a dataset of 2481 X-rays of 20 body parts + others, annotated in a multilabel fashion by radiology residents. Images are in DICOM format and Labels are categorical in csv file: Abdomen = 0, Ankle = 1, Cervical Spine = 2, Chest = 3, Clavicles = 4, Elbow = 5, Feet = 6, Finger = 7, Forearm = 8, Hand = 9, Hip = 10, Knee = 11, Lower Leg = 12, Lumbar Spine = 13, Others = 14, Pelvis = 15, Shoulder = 16, Sinus = 17, Skull = 18, Thigh = 19, Thoracic Spine = 20, Wrist = 21",
  "Dataset Path": "xray/unifesp"
}
]

```

```

3834     "Dataset Name": "knee",
3835     "Dataset Description": "It has 1,650 high-quality digital X-ray
3836         images of knee joints with a metadata file.",
3837     "Dataset Path": "xray/knee"
3838   },
3839   {
3840     "Dataset Name": "c19_radiograph",
3841     "Dataset Description": "COVID-19, lung opacity, normal and viral
3842         pneumonia chest X-ray (CXR) images are arranged in different
3843         sub-folders.",
3844     "Dataset Path": "xray/c19_radiograph"
3845   },
3846   {
3847     "Dataset Name": "simple_vs_community",
3848     "Dataset Description": "Bone Fracture X-ray Dataset: Simple vs.
3849         Comminuted Fractures organized in different subfolders",
3850     "Dataset Path": "xray/simple_vs_community"
3851   },
3852   {
3853     "Dataset Name": "nih_bbox",
3854     "Dataset Description": "This NIH Chest X-ray Dataset is comprised
3855         of 112,120 X-ray images with disease labels from 30,805
3856         unique patients. It has images in the image folder along with
3857         a label.csv with Class labels: 8 classes - Infiltrate,
3858         Atelectasis, Pneumonia, Cardiomegaly, Effusion, Pneumothorax,
3859         Mass, Nodule.",
3860     "Dataset Path": "xray/nih_bbox"
3861   },
3862   {
3863     "Dataset Name": "bone_break",
3864     "Dataset Description": "The dataset covers a range of bone
3865         fracture classes, such as avulsion fractures, comminuted
3866         fractures, fracture-dislocations, greenstick fractures,
3867         hairline fractures, impacted fractures, longitudinal
3868         fractures, oblique fractures, pathological fractures, and
3869         spiral fractures organized in separate subfolders",
3870     "Dataset Path": "xray/bone_break"
3871   },
3872   {
3873     "Dataset Name": "cov19_normal",
3874     "Dataset Description": "This dataset comprises a total of 800
3875         high-quality chest X-ray images, with 400 images depicting
3876         COVID-19 infected patients and the other 400 images
3877         representing normal cases (i.e., non-COVID patients) arranged
3878         in separate sub-folders.",
3879     "Dataset Path": "xray/cov19_normal"
3880   },
3881   {
3882     "Dataset Name": "dental",
3883     "Dataset Description": "Dental radiographs along with labels in
3884         csv files",
3885     "Dataset Path": "xray/dental"
3886   },
3887   {
3888     "Dataset Name": "bone_frac_small",
3889     "Dataset Description": "This dataset is designed for developing
3890         machine learning models for bone fracture classification and
3891         localization in tibia and fibula bones. It contains X-ray
3892         images in .PNG format along with labels in csv file",
3893     "Dataset Path": "xray/bone_frac_small"
3894   },
3895   {
3896     "Dataset Name": "knee_osteoporosis",
3897

```

```

3888
3889     "Dataset Description": "This knee XRay dataset contains 3 classes
3890         : normal, Osteopenia ,and Osteoporosis arranged in separate
3891         subfolders",
3892     "Dataset Path": "xray/knee_osteoporosis"
3893 },
3894 {
3895     "Dataset Name": "RNSA_pneumonia",
3896     "Dataset Description": "This dataset is a pre-processed version
3897         of the RSNA Pneumonia Detection Challenge dataset in PNG
3898         format along with the associated bounding box annotations as
3899         mask images. The metadata, including the patient information
3900         and bounding box coordinates, has been extracted and saved in
3901         CSV format.",
3902     "Dataset Path": "xray/RNSA_pneumonia"
3903 },
3904 {
3905     "Dataset Name": "8_object_detection",
3906     "Dataset Description": "Overview: The Chest X-ray 8 Subset
3907         dataset is a curated collection of chest radiographs for
3908         object detection models on thoracic diseases, with 790 images
3909         and 984 annotated bounding boxes in YOLO and Pascal VOC
3910         formats for diverse ML frameworks. Classes and Labels
3911         contained in associated csv file: 14 thoracic disease classes
3912         including Atelectasis, Cardiomegaly, Effusion, Infiltrate,
3913         Nodule, Mass, Pneumonia, Pneumothorax.",
3914     "Dataset Path": "xray/8_object_detection"
3915 },
3916 {
3917     "Dataset Name": "HBFMID",
3918     "Dataset Description": "Human Bone Fractures Multi-modal Image
3919         Dataset (HBFMID) is a collection of 1539 annotated medical
3920         images (X-ray and MRI) covering bone fractures in various
3921         locations (elbow, finger, forearm, humerus, shoulder, femur,
3922         shinbone, knee, hipbone, wrist, spinal cord, and some healthy
3923         bones) contained in the Image folder along with associated
3924         csv file",
3925     "Dataset Path": "xray/HBFMID"
3926 },
3927 {
3928     "Dataset Name": "FracAtlas",
3929     "Dataset Description": "It is a dataset of more than 14,000 X-Ray
3930         scans for classification, localization and segmentation of
3931         bone fractures. All the scans are available in JPG format
3932         along with proper annotations in separate sub-folders",
3933     "Dataset Path": "xray/FracAtlas"
3934 },
3935 {
3936     "Dataset Name": "pneumonia",
3937     "Dataset Description": "There are 5,863 X-Ray images (JPEG) and 2
3938         categories (Pneumonia/Normal) arranged in separate sub-
3939         folders",
3940     "Dataset Path": "xray/pneumonia"
3941 },
3942 {
3943     "Dataset Name": "pax_ray",
3944     "Dataset Description": "The PAX-Ray++ Dataset is a high-quality
3945         dataset designed to facilitate segmentation tasks for
3946         anatomical structures in chest radiographs available in Image
3947         subfolder and annotations in mask subfolder.",
3948     "Dataset Path": "xray/pax_ray"
3949 },
3950 {
3951     "Dataset Name": "lung_segmentation",
3952

```

```

3942     "Dataset Description": "This dataset contains over 500 x-ray
3943         scans with clinical labels collected by radiologists
3944             available in separate subfolders.",
3945     "Dataset Path": "xray/lung_segmentation"
3946 },
3947 {
3948     "Dataset Name": "shadow",
3949     "Dataset Description": "Normal Chest X-ray images and Bone Shadow
3950         images along with csv file.",
3951     "Dataset Path": "xray/shadow"
3952 },
3953 {
3954     "Dataset Name": "Angiography",
3955     "Dataset Description": "The ARCADE dataset (Automatic Region-
3956         based Coronary Artery Disease Diagnostics using X-ray
3957             Angiography) is organized into two task-specific directories
3958                 ('Task_Syntax_Segmentation' and 'Task_Stenosis_Segmentation')
3959                     , each containing flattened 'Images/' and 'masks/' subfolders
3960                         .",
3961     "Dataset Path": "xray/Angiography"
3962 },
3963 {
3964     "Dataset Name": "dental_panoramic",
3965     "Dataset Description": "Dental Disease Panoramic Dataset with
3966         segmentations on 31 classes: Classes: 0: Caries, 1: Crown, 2:
3967             Filling, 3: Implant, 4: Malaligned, 5: Mandibular Canal, 6:
3968                 Missing teeth, 7: Periapical lesion, 8: Retained root, 9:
3969                     Root Canal Treatment, 10: Root Piece, 11: Impacted tooth, 12:
3970                         Maxillary sinus, 13: Bone Loss, 14: Fracture teeth, 15:
3971                             Permanent Teeth, 16: Supra Eruption, 17: TAD, 18: Abutment,
3972                                 19: Attrition, 20: Bone defect, 21: Gingival former, 22:
3973                                     Metal band, 23: Orthodontic brackets, 24: Permanent retainer,
3974                                         25: Post-core, 26: Plating, 27: Wire, 28: Cyst, 29: Root
3975                                             resorption, 30: Primary teeth organized as different sub-
3976                                                 folders",
3977     "Dataset Path": "xray/dental_panoramic"
3978 },
3979 {
3980     "Dataset Name": "ALHI",
3981     "Dataset Description": "All images include a stem and a cup of
3982         the hip implant, and the images have to be X-ray images along
3983             with csv file containing metadata.",
3984     "Dataset Path": "xray/ALHI"
3985 },
3986 {
3987     "Dataset Name": "humerus_fractures",
3988     "Dataset Description": "Deep Learning-Driven Diagnosis of Humerus
3989         Fractures from Radiographic Data. Images contain x-ray
3990             images of humerus fractures and non-fractures in separate
3991                 subfolders.",
3992     "Dataset Path": "xray/humerus_fractures"
3993 },
3994 {
3995     "Dataset Name": "multiclass_knee_osteoporosis",
3996     "Dataset Description": "The dataset is divided into three primary
3997         categories: (1) Normal: Images of knees with no signs of
3998             osteoporosis., (2) Osteopenia: Images showing early stages of
3999                 bone density loss, and (3) Osteoporosis: Images indicating
4000                     advanced bone density degradation organized as different
4001                         subfolders",
4002     "Dataset Path": "xray/multiclass_knee_osteoporosis"
4003 },
4004 {
4005     "Dataset Name": "rsna-breast-cancer-detection",

```

```

3996
3997     "Dataset Description": "Region of Interests extracted from breast
3998         X-ray images. There are no labels, just .png images.",
3999     "Dataset Path": "xray/rsna-breast-cancer-detection"
4000   },
4001   {
4002     "Dataset Name": "RANZCR",
4003     "Dataset Description": "For detecting the presence and position
4004         of catheters and lines on chest x-rays. The .csv file
4005         contains image IDs, binary labels, and patient IDs with
4006         columns: Columns: StudyInstanceUID (unique ID for each image)
4007         , ETT - Abnormal (endotracheal tube placement abnormal), ETT
4008         - Borderline (borderline abnormal), ETT - Normal (normal),
4009         NGT - Abnormal (nasogastric tube placement abnormal), NGT -
4010         Borderline (borderline abnormal), NGT - Incompletely Imaged (
4011         inconclusive due to imaging), NGT - Normal (normal), CVC -
4012         Abnormal (central venous catheter placement abnormal), CVC -
4013         Borderline (borderline abnormal), CVC - Normal (normal), Swan
4014         Ganz Catheter Present, PatientID (unique ID for each patient
4015         ).",
4016     "Dataset Path": "xray/RANZCR"
4017   },
4018   {
4019     "Dataset Name": "FractureFusion",
4020     "Dataset Description": "From avulsion fractures to spiral
4021         fractures, this dataset is a rich repository of diverse cases
4022         , including comminuted fractures, fracture-dislocations,
4023         greenstick fractures, hairline fractures, impacted fractures,
4024         longitudinal fractures, oblique fractures, pathological
4025         fractures arranged as different subfolders",
4026     "Dataset Path": "xray/FractureFusion"
4027   },
4028   {
4029     "Dataset Name": "HeelBone",
4030     "Dataset Description": "Heel Bone X-Ray Dataset consists of 3,956
4031         X-ray images of the foot, primarily focused on detecting and
4032         classifying heel bone diseases with annotations arranged in
4033         label.csv",
4034     "Dataset Path": "xray/HeelBone"
4035   }
4036 ]
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049

```

Listing 8: Dataset Descriptions for Histopathology Modality

```

4032 [
4033   {
4034     "Dataset Name": "breast_hist",
4035     "Dataset Description": "Breast Histopathology Images with
4036         Invasive Ductal Carcinoma (IDC). There's no labels for this
4037         dataset, only images.",
4038     "Dataset Path": "histopathology/breast_hist"
4039   },
4040   {
4041     "Dataset Name": "BreaKHis_400X",
4042     "Dataset
4043         Description": "Breast cancer images on histopathology slides.
4044         The BreaKHis database contains microscopic biopsy images
4045         benign and malignant breast tumors in separate subfolders.",
4046     "Dataset Path": "histopathology/BreaKHis_400X"
4047   },
4048   {
4049     "Dataset Name": "lung_and_colon",
4050     "Dataset Description": "Lung and Colon Cancer Histopathological
4051         Images: 25000 images of 5 classes: Lung benign tissue, Lung
4052         adenocarcinoma, Lung squamous cell carcinoma, Colon
4053         adenocarcinoma, Colon benign tissue in separate subfolders."
4054   }
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4590
4591
4592
4593
4594
4595
4596
4597
4598
4599
4600
4601
4602
4603
4604
4605
4606
4607
4608
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666
4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871
4872
4873
4874
4875
4876
4877
4878
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899
4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000
5001
5002
5003
5004
5005
5006
5007
5008
5009
5010
5011
5012
5013
5014
5015
5016
5017
5018
5019
5020
5021
5022
5023
5024
5025
5026
5027
5028
5029
5030
5031
5032
5033
5034
5035
5036
5037
5038
5039
5040
5041
5042
5043
5044
5045
5046
5047
5048
5049
5050
5051
5052
5053
5054
5055
5056
5057
5058
5059
5060
5061
5062
5063
5064
5065
5066
5067
5068
5069
5070
5071
5072
5073
5074
5075
5076
5077
5078
5079
5080
5081
5082
5083
5084
5085
5086
5087
5088
5089
5090
5091
5092
5093
5094
5095
5096
5097
5098
5099
5100
5101
5102
5103
5104
5105
5106
5107
5108
5109
5110
5111
5112
5113
5114
5115
5116
5117
5118
5119
5120
5121
5122
5123
5124
5125
5126
5127
5128
5129
5130
5131
5132
5133
5134
5135
5136
5137
5138
5139
5140
5141
5142
5143
5144
5145
5146
5147
5148
5149
5150
5151
5152
5153
5154
5155
5156
5157
5158
5159
5160
5161
5162
5163
5164
5165
5166
5167
5168
5169
5170
5171
5172
5173
5174
5175
5176
5177
5178
5179
5180
5181
5182
5183
5184
5185
5186
5187
5188
5189
5190
5191
5192
5193
5194
5195
5196
5197
5198
5199
5200
5201
5202
5203
5204
5205
5206
5207
5208
5209
5210
5211
5212
5213
5214
5215
5216
5217
5218
5219
5220
5221
5222
5223
5224
5225
5226
5227
5228
5229
5230
5231
5232
5233
5234
5235
5236
5237
5238
5239
5240
5241
5242
5243
5244
5245
5246
5247
5248
5249
5250
5251
5252
5253
5254
5255
5256
5257
5258
5259
5260
5261
5262
5263
5264
5265
5266
5267
5268
5269
5270
5271
5272
5273
5274
5275
5276
5277
5278
5279
5280
5281
5282
5283
5284
5285
5286
5287
5288
5289
5290
5291
5292
5293
5294
5295
5296
5297
5298
5299
5300
5301
5302
5303
5304
5305
5306
5307
5308
5309
5310
5311
5312
5313
5314
5315
5316
5317
5318
5319
5320
5321
5322
5323
5324
5325
5326
5327
5328
5329
5330
5331
5332
5333
5334
5335
5336
5337
5338
5339
5340
5341
5342
5343
5344
5345
5346
5347
5348
5349
5350
5351
5352
5353
5354
5355
5356
5357
5358
5359
5360
5361
5362
5363
5364
5365
5366
5367
5368
5369
5370
5371
5372
5373
5374
5375
5376
5377
5378
5379
5380
5381
5382
5383
5384
5385
5386
5387
5388
5389
5390
5391
5392
5393
5394
5395
5396
5397
5398
5399
5400
5401
5402
5403
5404
5405
5406
5407
5408
5409
5410
5411
5412
5413
5414
5415
5416
5417
5418
5419
5420
5421
5422
5423
5424
5425
5426
5427
5428
5429
5430
5431
5432
5433
5434
5435
5436
5437
5438
5439
5440
5441
5442
5443
5444
5445
5446
5447
5448
5449
5450
5451
5452
5453
5454
5455
5456
5457
5458
5459
5460
5461
5462
5463
5464
5465
5466
5467
5468
5469
5470
5471
5472
5473
5474
5475
5476
5477
5478
5479
5480
5481
5482
5483
5484
5485
5486
5487
5488
5489
5490
5491
5492
5493
5494
5495
5496
5497
5498
5499
5500
5501
5502
5503
5504
5505
5506
5507
5508
5509
5510
5511
5512
5513
5514
5515
5516
5517
5518
5519
5520
5521
5522
5523
5524
5525
5526
5527
5528
5529
5530
5531
5532
5533
5534
5535
5536
5537
5538
5539
5540
5541
5542
5543
5544
5545
5546
5547
5548
5549
5550
5551
5552
5553
5554
5555
5556
5557
5558
5559
5560
5561
5562
5563
5564
5565
5566
5567
5568
5569
5570
5571
5572
5573
5574
5575
5576
5577
5578
5579
5580
5581
5582
5583
5584
5585
5586
5587
5588
5589
5590
5591
5592
5593
5594
5595
5596
5597
5598
5599
5600
5601
5602
5603
5604
5605
5606
5607
5608
5609
5610
5611
5612
5613
5614
5615
5616
5617
5618
5619
5620
5621
5622
5623
5624
5625
5626
5627
5628
5629
5630
5631
5632
5633
5634
5635
5636
5637
5638
5639
5640
5641
5642
5643
5644
5645
5646
5647
5648
5649
5650
5651
5652
5653
5654
5655
5656
5657
5658
5659
5660
5661
5662
5663
5664
5665
5666
5667
5668
5669
5670
5671
5672
5673
5674
5675
5676
5677
5678
5679
5680
5681
5682
5683
5684
5685
5686
5687
5688
5689
5690
5691
5692
5693
5694
5695
5696
5697
5698
5699
5700
5701
5702
5703
5704
5705
5706
5707
5708
5709
5710
5711
5712
5713
5714
5715
5716
5717
5718
5719
5720
5721
5722
5723
5724
5725
5726
5727
5728
5729
5730
5731
5732
5733
5734
5735
5736
5737
5738
5739
5740
5741
5742
5743
5744
5745
5746
5747
5748
5749
5750
5751
5752
5753
5754
5755
5756
5757
5758
5759
5760
5761
5762
5763
5764
5765
5766
5767
5768
5769
5770
5771
5772
5773
5774
5775
5776
5777
5778
5779
5780
5781
5782
5783
5784
5785
5786
5787
5788
5789
5790
5791
5792
5793
5794
5795
5796
5797
5798
5799
5800
5801
5802
5803
5804
5805
5806
5807
5808
5809
5810
5811
5812
5813
5814
5815
5816
5817
5818
5819
5820
5821
5822
5823
5824
5825
5826
5827
5828
5829
5830
5831
5832
5833
5834
5835
5836
5837
5838
5839
5840
5841
5842
5843
5844
5845
5846
5847
5848
5849
5850
5851
5852
5853
5854
5855
5856
5857
5858
5859
5860
5861
5862
5863
5864
5865
5866
5867
5868
5869
5870
5871
5872
5873
5874
5875
5876
5877
5878
5879
5880
5881
5882
5883
5884
5885
5886
5887
5888
5889
5890
5891
5892
5893
5894
5895
5896
5897
5898
5899
5900
5901
5902
5903
5904
5905
5906
5907
5908
5909
5910
5911
5912
5913
5914
5915
5916
5917
5918
5919
5920
5921
5922
5923
5924
5925
5926
5927
5928
5929
5930
5931
5932
5933
5934
5935
5936
5937
5938
5939
5940
5941
5942
5943
5944
5945
5946
5947
5948
5949
5950
5951
5952
5953
5954
5955
5956
5957
5958
5959
5960
5961
5962
5963
5964
5965
5966
5967
5968
5969
5970
5971
5972
5973
5974
5975
5976
5977
5978
5979
5980
5981
5982
5983
5984
5985
5986
5987
5988
5989
5990
5991
5992
5993
5994
5995
5996
5997
5998
5999
6000
6001
6002
6003
6004
6005
6006
6007
6008
6009
6010
6011
6012
6013
6014
6015
6016
6017
6018
6019
6020
6021
6022
6023
6024
6025
6026
6027
6028
6029
6030
6031
6032
6033
6034
6035
6036
6037
6038
6039
6040
6041
6042
6043
6044
6045
6046
6047
6048
6049
6050
6051
6052
6053
6054
6055
6056
6057
6058
6059
6060
6061
6062
6063
6064
6065
6066
6067
6068
6069
6070
6071
6072
6073
6074
6075
6076
6077
6078
6079
6080
6081
6082
6083
6084
6085
6086
6087
6088
6089
6090
6091
6092
6093
6094
6095
6096
6097
6098
6099
6100
6101
6102
6103

```

```

4050     "Dataset Path": "histopathology/lung_and_colon"
4051   },
4052   {
4053     "Dataset Name": "gastric_cancer",
4054     "Dataset Description": "Gastric Cancer Histopathology Tissue
4055       Image Dataset focuses on the tumor microenvironment (TME) and
4056       includes images categorized into eight distinct tissue types
4057       : ADI: Adipose (fat tissue), BACK: Background (non-tissue
4058       areas), DEB: Debris (cellular waste), LYM: Lymphocytes (
4059       immune cells), MUC: Mucus (protective secretion), MUS: Smooth
4060       Muscle (muscle tissue), NORM: Normal Colon Mucosa (healthy
4061       tissue for reference), STR: Cancer-associated Stroma (
4062       connective tissue around the tumor), TUM: Tumor (cancerous
4063       tissue) - all arranged in different subfolders. ",
4064     "Dataset Path": "histopathology/gastric_cancer"
4065   },
4066   {
4067     "Dataset Name": "gastro_cancer_msi_vs_mss",
4068     "Dataset Description": "The dataset contains histological images
4069       for MSI vs MSS classification in gastrointestinal cancer
4070       arranged in different sub-folders.",
4071     "Dataset Path": "histopathology/gastro_cancer_msi_vs_mss"
4072   },
4073   {
4074     "Dataset Name": "breast_cancer_segmentation",
4075     "Dataset Description": "Breast Cancer Cell Segmentation with
4076       corresponding images and masks in separate subfolders.",
4077     "Dataset Path": "histopathology/breast_cancer_segmentation"
4078   },
4079   {
4080     "Dataset Name": "ovarian_cancer",
4081     "Dataset Description": "Ovarian Cancer & Subtypes Dataset
4082       Histopathology: This dataset includes histopathology images
4083       of 4 subtypes of Ovarian cancer and also non cancerous
4084       histopathological images organized in separate subfolders",
4085     "Dataset Path": "histopathology/ovarian_cancer"
4086   },
4087   {
4088     "Dataset Name": "breast_cancer_histo",
4089     "Dataset Description": "breast cancer histopathology. JPG images
4090       with classifications benign or malignant organized as
4091       separate subfolders",
4092     "Dataset Path": "histopathology/breast_cancer_histo"
4093   },
4094   {
4095     "Dataset Name": "BreCaHAD",
4096     "Dataset Description": "a dataset for breast cancer histopathological
4097       annotation and diagnosis with images belonging to six classes
4098       , namely mitosis, apoptosis, tumor nuclei, non-tumor nuclei,
4099       tubule, and non-tubule arranged in separate subfolders",
4100     "Dataset Path": "histopathology/BreCaHAD"
4101   },
4102   {
4103     "Dataset Name": "melanoma",
4104     "Dataset Description": "This dataset is a melanoma specific
4105       dataset with nuclei and tissue annotations along with
4106       original images in separate subfolders.",
4107     "Dataset Path": "histopathology/melanoma"
4108   },
4109   {
4110     "Dataset Name": "choledoch",
4111     "Dataset Description": "This is a database for both microscopy
4112       hyperspectral and color images of cholangiocarcinoma,
4113       including 880 scenes among which 689 scenes are samples with

```

```

4104     part of cancer areas (L), 49 scenes full of cancer areas (N),
4105     and 142 scenes without cancer areas (P) organized as
4106     separate subfolders",
4107     "Dataset Path": "histopathology/choledoch"
4108   },
4109   {
4110     "Dataset Name": "histopath-sn",
4111     "Dataset Description": "This is a Kaggle dataset, with the task
4112       to classify patches: Bronchus and lung samples in image
4113       folder along with labels in separate csv file.",
4114     "Dataset Path": "histopathology/histopath-sn"
4115   },
4116   {
4117     "Dataset Name": "ULMS",
4118     "Dataset Description": "Uterine leiomyosarcoma (ULMS) dataset
4119       comprises mitosis count, necrosis, and nuclear atypia with
4120       labels in separate csv file",
4121     "Dataset Path": "histopathology/ULMS"
4122   },
4123   {
4124     "Dataset Name": "MonuSeg",
4125     "Dataset Description": "The dataset comprises nuclei from seven
4126       organs with associated annotations in csv file.",
4127     "Dataset Path": "histopathology/MonuSeg"
4128   },
4129   {
4130     "Dataset Name": "kmc_kidney",
4131     "Dataset Description": "The introduced KMC kidney histopathology
4132       dataset includes non-cancerous (Grade-0) and cancerous (Grade
4133       -1 to Grade-4) images of the Renal Clear Cell Carcinoma
4134       organized as separate subfolders",
4135     "Dataset Path": "histopathology/kmc_kidney"
4136   },
4137   {
4138     "Dataset Name": "histo-img-text",
4139     "Dataset Description": "This is a kaggle dataset with
4140       histopathology image-text pairs",
4141     "Dataset Path": "histopathology/histo-img-text"
4142   },
4143   {
4144     "Dataset Name": "cellnet",
4145     "Dataset Description": "CellNet is a meticulously curated dataset
4146       featuring over 120,000 high-quality medical images
4147       representing over 20 organ/cancer classes arranged as
4148       different subfolders. ",
4149     "Dataset Path": "histopathology/cellnet"
4150   },
4151   {
4152     "Dataset Name": "PanNuke",
4153     "Dataset Description": "Nuclei instance segmentation and
4154       classification dataset with exhaustive nuclei labels across
4155       19 different tissue types. In total the dataset contains
4156       205,343 labeled nuclei, each with an instance segmentation
4157       mask in separate datasets.",
4158     "Dataset Path": "histopathology/PanNuke"
4159   },
4160   {
4161     "Dataset Name": "NPC-88k-Public",
4162     "Dataset Description": "88k histopathology patches of normal,
4163       lymphoid hyperplasia (LHP), nasopharyngeal inflammation (NPI)
4164       , and nasopharyngeal carcinoma (NPC) organized in separate
4165       subfolders.",
4166     "Dataset Path": "histopathology/NPC-88k-Public"
4167   },

```

```

4158
4159     {
4160         "Dataset Name": "EBHI",
4161         "Dataset
4162             Description": "The dataset encompasses various categories,
4163             including normal (76 images and 76 ground truth images),
4164             polyp (474 images and 474 ground truth images), low-grade
4165             intraepithelial neoplasia (639 images and 639 ground truth
4166             images), high-grade intraepithelial neoplasia (186 images and
4167             186 ground truth images), serrated adenoma (58 images and 58
4168             ground truth images), and adenocarcinoma (795 images and 795
4169             ground truth images) arranged in different subfolders",
4170         "Dataset Path": "histopathology/EBHI"
4171     }
4172 ]

```

4171 C.3 DETECTING AND ADDRESSING DATA QUALITY ISSUES FOR DATA PRE-PROCESSING 4172 AGENT

4173 One of the primary steps in data pre-processing involves identifying data quality issues and removing
4174 samples that negatively impact the overall data quality. In this work, we address three key data quality
4175 issues *viz.* **off-topic samples**, **near duplicates**, and **label errors** (Gröger et al., 2025; 2024; 2023)
4176 each of which can significantly compromise the reliability of machine learning models, particularly
4177 in sensitive domains like medical imaging.

- 4178 • **Off-topic samples** refer to irrelevant inputs mistakenly included in the dataset (e.g., from
4179 unrelated modalities or corrupted acquisitions). These introduce noise, distort evaluation
4180 metrics, and hinder model convergence.
- 4181 • **Near duplicates** are different views of the same object, including exact copies. Their
4182 presence artificially reduces the diversity of the training set, introduces redundancy, and may
4183 lead to data leakage between training and evaluation sets.
- 4184 • **Label errors** are incorrectly annotated examples that can misguide both model training and
4185 evaluation, leading to degraded performance and spurious generalization.

4186 The dataset is formalized as $\mathcal{X} = \{(x_i, l_i) \mid i \in \mathcal{I}\}$, where each x_i is a sample, l_i is its label among
4187 L classes, and $\mathcal{I} = \{1, \dots, N\}$ the index set. For each issue type, a scoring function $s(\cdot)$ is defined
4188 that maps individual samples or sample pairs to a score in $[0, 1]$, where lower values indicate higher
4189 likelihood of an issue. Ranking the samples by these scores yields a prioritized list for inspection or
4190 automated filtering based on a pre-defined threshold.

4191 REPRESENTATION LEARNING

4192 A deep feature extractor $f(\cdot; \theta)$ was trained using self-supervised learning (SSL) methods (*SimCLR*
4193 or *DINO*), both of which were implemented with a Vision Transformer (ViT) backbone. Each sample
4194 x_i was embedded into a latent space as $e_i = f(x_i; \theta) \in \mathbb{R}^D$, where D denotes the feature dimension.
4195 To ensure consistent geometry across methods, ℓ_2 -normalization was applied so that all embeddings
4196 lie on a unit hypersphere.

4197 Cosine similarity was adopted to define the distance metric:

$$4198 \text{sim}(e_i, e_j) = \frac{e_i^\top e_j}{\|e_i\|_2 \|e_j\|_2}, \quad \text{dist}(e_i, e_j) = \frac{1 - \text{sim}(e_i, e_j)}{2}.$$

4200 ISSUE-SPECIFIC DETECTION STRATEGIES

4201 **Off-topic Detection.** Off-topic samples were identified using agglomerative clustering with single
4202 linkage in the representation space. The merging behavior of clusters was analyzed, and samples that
4203 were merged at higher distances or at later stages with larger clusters were considered more likely to
4204 be anomalous. A scoring function $sot(e_i)$ was constructed based on merge depth and inter-cluster
4205 distance dynamics.

4212 **Near Duplicate Detection.** Candidate near-duplicate pairs were detected by evaluating pairwise
 4213 distances between all sample embeddings. A simple ranking function was applied:
 4214

$$s_{\text{ND}}(e_i, e_j) = \text{dist}(e_i, e_j),$$

4215 where smaller distances were interpreted as a higher likelihood of duplication.
 4216

4218 **Label Error Detection.** Label errors were inferred based on a ratio between intra-class and inter-
 4219 class distances. For each sample e_i , the following definitions were used:
 4220

$$m_{=} = \min_{j \in \mathcal{I}, l_j = l_i} \text{dist}(e_i, e_j), \quad m_{\neq} = \min_{j \in \mathcal{I}, l_j \neq l_i} \text{dist}(e_i, e_j),$$

$$s_{\text{LE}}(e_i) = \frac{m_{\neq}^2(e_i)}{m_{=}^2(e_i) + m_{\neq}^2(e_i)}.$$

4225 Lower scores were interpreted as indicating a higher likelihood of mislabeling, particularly when the
 4226 nearest neighbor belonged to a different class.
 4227

4228 In all three cases, the local structure of the embedding space was leveraged by the cleaning function
 4229 used in Tool 9 of the Listing 1. Cluster distances were evaluated using only the nearest neighbors
 4230 for off-topic detection, proximity among sample pairs was assessed for duplicate identification, and
 4231 comparative distances to same- and different-class neighbors were exploited to detect label errors.
 4232

4233 C.4 COLLECTION OF FEDERATED LEARNING ALGORITHMS

4234 Federated Learning (FL) has evolved significantly beyond its initial formulation of model averaging,
 4235 with numerous algorithmic innovations developed to address practical challenges such as data
 4236 heterogeneity, personalization, privacy preservation, and limited client resources (McMahan et al.,
 4237 2017; Tan & Wang; Tan et al., 2023). In this work, we utilize a set of **40** key federated learning
 4238 (FL) algorithms, covering core, personalized, generalizable, and adaptive methods, as summarized
 4239 in Tables 2-4. The algorithm description required by server-based federated training agents for FL
 4240 algorithm selection is provided in Listing 9.

4241 The selected algorithms reflect the diversity and progression of research in FL across three main axes:
 4242

4243 1. **Foundational and General-Purpose Methods:**

4244 We begin with core algorithms such as *FedAvg*, *FedAvgM*, and *FedProx*, which establish
 4245 the baseline principles of client-server aggregation and account for statistical and system
 4246 heterogeneity. These methods are essential for benchmarking and provide the backbone
 4247 upon which many subsequent algorithms are built.

4248 2. **Personalization-Oriented Methods:**

4249 Recognizing the need to adapt to non-IID data across clients, we include algorithms like
 4250 *FedRep*, *FedPer*, *Ditto*, *pFedHN*, and *Per-FedAvg*. These approaches personalize part of the
 4251 model (e.g., classifier heads or entire layers), use meta-learning, or leverage client-specific
 4252 adaptation strategies. Methods such as *pFedMe* and *FedEM* extend this personalization
 4253 through bi-level optimization and mixture modeling, respectively.

4254 3. **Robustness, Adaptivity, and Generalization:**

4255 To tackle challenges of out-of-distribution generalization and domain shifts, we incorporate
 4256 algorithms like *FedIRR*, *FedSR*, and *ADCOL*, which emphasize invariant representation
 4257 learning and adversarial feature alignment. Techniques such as *FedDyn*, *FedFomo*, and
 4258 *FedRoD* introduce dynamic regularization and adaptive weighting to stabilize optimization
 4259 in heterogeneous environments. Moreover, algorithms like *FedBN* and *FedAP* address
 4260 domain-specific normalization challenges, particularly in healthcare contexts.

4261 4. **Emerging and Specialized Directions:**

4262 The inclusion of recent methods such as *Floco*, *FedAS*, and *PeFLL* highlights advancements
 4263 in adaptive aggregation, inter-client relationship modeling, and meta-learned personalization.
 4264 Additionally, *MOON*, *FedGen*, and *CCVR* represent innovative uses of contrastive learning,
 4265 data-free distillation, and virtual representation calibration.

4266 The rationale for selecting this curated list is threefold:

4266 • **Comprehensiveness:** The algorithms span from classic to state-of-the-art methods, ensuring
 4267 broad coverage of the field.
 4268 • **Modular Design Potential:** These algorithms are suitable for integration into modular
 4269 federated learning pipelines, facilitating agent-based automation and tool invocation.
 4270 • **Relevance to Real-World Scenarios:** Many chosen methods address constraints encoun-
 4271 tered in practical deployments, including label imbalance, resource limitations, domain
 4272 adaptation, and personalization needs.
 4273

4274 This comprehensive collection enables systematic benchmarking, comparative evaluation, and modu-
 4275 lar composition in our federated learning framework *FedAgentBench*. Each method contributes unique
 4276 strengths and trade-offs, making them valuable candidates for real-world and research applications.
 4277

4278 **Listing 9: Federated Learning Algorithm Descriptions for Server-based algorithm selector agents**

```

4279   [
4280     {
4281       "algorithm": "FedAvg",
4282       "description": "The foundational algorithm in federated
4283           learning, where clients perform multiple steps of local
4284           stochastic gradient descent (SGD) and periodically average
4285           their models on a central server. It is simple and
4286           communication-efficient but struggles with non-IID data
4287           distributions."
4288     },
4289     {
4290       "algorithm": "FedAvgM",
4291       "description": "An extension of FedAvg that integrates server-
4292           side momentum during model aggregation. This is a classical
4293           federated learning approach that stabilizes training and
4294           improves convergence in the presence of data heterogeneity
4295           across clients."
4296     },
4297     {
4298       "algorithm": "FedProx",
4299       "description": "Classical federated learning algorithm that
4300           enhances FedAvg by adding a proximal term to the local
4301           objective functions, discouraging local updates from drifting
4302           too far from the global model. This addresses system and
4303           statistical heterogeneity among clients."
4304     },
4305     {
4306       "algorithm": "SCAFFOLD",
4307       "description": "Classical federated learning algorithm that
4308           incorporates control variates to correct client-drift caused
4309           by non-IID data. Each client maintains local control
4310           variables to align updates with the global objective,
4311           improving convergence stability."
4312     },
4313     {
4314       "algorithm": "MOON",
4315       "description": "Traditional Federated learning algorithm that
4316           implements model-level contrastive learning by aligning
4317           current local models with the global model while contrasting
4318           them with past local models. This enhances representation
4319           learning under non-IID settings."
4320     },
4321     {
4322       "algorithm": "FedDyn",
4323       "description": "Regularization-based federated learning
4324           approach that introduces a dynamic regularization term into
4325           local objectives that evolves over time to better match the
4326           global objective. This mechanism helps mitigate divergence
4327           and stabilizes training in heterogeneous environments."
4328     },
4329   ],

```

Table 4: Overview of Federated Learning Algorithms (Part 1)

Method	Source	Key Idea	Strengths	Limitations
FedAvg (McMahan et al., 2017)	McMahan et al., 2016	Clients perform local SGD and periodically average with the server.	Simple and communication-efficient.	Degrades with non-IID data due to client drift.
FedAvgM (Hsu et al., 2019)	Hsu et al., 2019	Adds server-side momentum to FedAvg.	Improves convergence on non-IID data.	Requires careful momentum tuning.
FedMD (Li & Wang, 2019)	Li et al., NeurIPS 2019	Uses public dataset for knowledge distillation across heterogeneous models.	Supports diverse architectures.	Requires public dataset.
FedPer (Arivazhagan et al., 2019)	Arivazhagan et al., arXiv 2019	Uses client-specific layers with shared global layers.	Balances global and local learning.	Designing layer split is non-trivial.
LG-FedAvg (Liang et al., 2020)	Liang et al., NeurIPS 2019 Workshop	Aggregates global layers, retains local ones.	Preserves local personalization.	Complex model synchronization.
CFL (Sattler et al., 2019)	Sattler et al., arXiv 2019	Clusters clients and trains separate models.	Addresses data heterogeneity.	Doesn't scale well with many clusters.
FedProx (Li et al., 2020b)	Li et al., 2020	Adds proximal term to local loss.	Handles statistical/system heterogeneity.	May slow down convergence.
SCAFFOLD (Karimireddy et al., 2020)	Karimireddy et al., 2020	Uses control variates to correct drift.	Better convergence on non-IID data.	Extra storage and computation.
APFL (Deng et al., 2020)	Deng et al., arXiv 2020	Adaptive mixing of global and local models.	Combines generalization and personalization.	Requires careful mixing parameter tuning.
Per-FedAvg (Fallah et al., 2020)	Fallah et al., NeurIPS 2020	Combines FL with MAML.	Enables fast personalization.	Needs second-order gradients.
pFedMe (Dinh et al., 2022)	Dinh et al., NeurIPS 2020	Uses Moreau envelopes for bi-level optimization.	Fast convergence and good personalization.	Requires tuning of regularization.
MOON (Li et al., 2021a)	Li et al., CVPR 2021	Aligns local and global models via contrastive loss.	Strong representation learning.	Needs previous model storage.
FedDyn (Acar et al., 2021)	Acar et al., ICLR 2021	Dynamic regularization to align objectives.	Mitigates client drift.	More complex optimization.
FedGen (Zhu et al., 2021)	Zhu et al., ICML 2021	Uses synthetic data for knowledge distillation.	Enables data-free generalization.	Depends on generator quality.
FedOpt (Reddi et al., 2021)	Reddi et al., ICLR 2021	Uses adaptive optimizers (Adam/Yogi) in FL.	Fast/stable convergence.	Hyperparameter tuning required.

4374

4375

4376

4377

4378

4379

Table 5: Overview of Federated Learning Algorithms (Part 2)

Method	Source	Key Idea	Strengths	Limitations
CCVR (Luo et al., 2021)	Wang et al., NeurIPS 2021	Virtual representations for calibration.	No real data sharing needed.	Relies on distribution approximations.
FedEM (Marfoq et al., 2022)	Marfoq et al., NeurIPS 2021	Mixture model for multi-task personalization.	Captures cross-client distributions.	Assumes shared latent structure.
Ditto (Li et al., 2021c)	Li et al., ICML 2021	Maintains global and personalized models.	Robust and fair personalization.	Needs dual model training.
FedRep (Collins et al., 2023)	Collins et al., ICML 2021	Shared encoder with local classifiers.	Combines global and local strengths.	Coordination needed for shared layer.
pFedHN (Shamsian et al., 2021)	Shamsian et al., ICML 2021	Hypernetworks generate personalized models.	Communication efficient.	Complex hypernetwork training.
FedFomo (Zhang et al., 2021)	Zhang et al., ICLR 2021	Aggregates based on client similarity.	Personalization without raw data.	Similarity computation overhead.
FedBN (Li et al., 2021d)	Li et al., ICLR 2021	Local BN layers for domain adaptation.	Improves performance on non-IID data.	No global BN normalization.
FedLC (Zhang et al., 2022)	Zhang et al., ICML 2022	Logits calibration to handle label skew.	Effective on imbalanced datasets.	Needs label distribution estimation.
MetaFed (Chen et al., 2023b)	IJCAI 2022	Cyclic knowledge distillation across federations.	Enhances collaboration.	Federation coordination required.
FedRoD (Chen & Chao, 2022)	ICLR 2022	Adaptive aggregation for balancing general/personal models.	Personalized and generalizable.	May fail under high heterogeneity.
FedProto (Tan et al., 2022)	AAAI 2022	Prototype-based feature alignment.	Preserves global semantics.	Quality depends on prototypes.
pFedLA (Ma et al., 2022)	Ma et al., CVPR 2022	Layer-wise model aggregation.	Fine-grained personalization.	Management complexity.
FedBABU (Oh et al., 2022)	Oh et al., ICLR 2022	Aggregates body and keeps local heads.	Improves representation learning.	Less consistent predictions.
FedAP (Lu et al., 2022)	Chen et al., IEEE 2022	Adaptive BN for healthcare FL.	Handles domain shift.	Sensitive to BN statistics.

4425

4426

4427

4428
4429
4430
4431
4432
4433
4434
4435

Table 6: Overview of Federated Learning Algorithms (Part 3)

Method	Source	Key Idea	Strengths	Limitations
FedSR (Nguyen et al., 2022a)	NeurIPS 2022	Domain generalization via representation regularization.	Lightweight and simple.	May fail in extreme domain shift.
FedALA (Zhang et al., 2023)	AAAI 2023	Adaptive local aggregation weights.	Relevance-aware updates.	Unstable weight estimation.
FedFed (Yang et al., 2023)	Yang et al., NeurIPS 2023	Distills critical features.	Improves generalization.	Needs good feature selection.
Elastic Aggrega- tion (Chen et al., 2023a)	Chen et al., CVPR 2023	Sensitivity-based update weighting.	Balances adaptation/stability.	Adds computation.
ADCOL (Li et al., 2023b)	ICML 2023	Adversarial alignment of features.	Handles domain shift.	Adversarial training instability.
FedIIR (Guo et al., 2023)	ICML 2023	Learns invariant relationships for OOD generalization.	Strong generalization.	Needs assumptions on invariance.
pFedSim (Tan et al., 2023)	Tan et al., arXiv 2023	Similarity-based aggregation.	Enables personalization.	Hard to measure similarity.
PeFLL (Scott et al., 2025)	ICLR 2024	Meta-learns to personalize clients.	Fast client adaptation.	High computation cost.
FLUTE (Liu et al., 2024a)	ICML 2024	Efficient rep learning under underparameterization.	Resource efficient.	May sacrifice expressivity.
FedAS (Yang et al., 2024)	CVPR 2024	Reduces global-local inconsistency.	More consistent updates.	More complex training.
Floco (Grinwald et al., 2025)	NeurIPS 2024	Uses connected modes to model clients.	Leverages inter-client structure.	Needs client connectivity info.

4476
4477
4478
4479
4480
4481

Table 7: Categorization of FL Algorithms

4482
4483

Category	Algorithms
(i) Classical FL algorithms	FedAvg, FedAvgM, FedProx, SCAFFOLD, MOON, FedLC
(ii) Personalized FL algorithms	Per-FedAvg, pFedMe, FedRep, FedPer, FedBN, pFedLA, pFedHN, FedFomo, LG-FedAvg, APFL, FedEM, pFedSim, FedBABU, CCVR
(iii) Regularization-based approaches	Ditto, FedDyn, FedRoD, FedAS, SCAFFOLD, pFedMe
(iv) Knowledge Distillation-based methods	FedGen, FedMD, FedFed, MetaFed
(v) Domain generalization techniques	FedSR, FedIIR, ADCOL, FedProto, FedAP
(vi) Optimization and scheduling variants	FedOpt, FedAvgM, FedALA, Elastic Aggregation, FLUTE, PeFLL, CFL

4497

4498

```

4499  {
4500      "algorithm": "FedLC",
4501      "description": "Classical federated learning algorithm that
4502          applies logits calibration techniques during local training
4503          to address label distribution skew. This helps balance
4504          prediction confidence and improve accuracy on imbalanced or
4505          non-IID datasets."
4506  },
4507  {
4508      "algorithm": "FedGen",
4509      "description": "Personalized Federated Learning leveraging
4510          knowledge distillation that uses a server-side generative
4511          model to synthesize data representations for knowledge
4512          distillation, enabling model personalization without
4513          requiring access to client data. This preserves privacy while
4514          supporting generalization."
4515  },
4516  {
4517      "algorithm": "CCVR",
4518      "description": "Personalized Federated Learning that uses
4519          virtual representations drawn from approximated data
4520          distributions to calibrate classifiers. This approach
4521          improves generalization in non-IID scenarios without needing
4522          to exchange actual data between clients."
4523  },
4524  {
4525      "algorithm": "FedOpt",
4526      "description": "Federated adaptive optimization scheme that
4527          extends FedAvg by integrating adaptive gradient methods like
4528          FedAdam, FedYogi, and FedAdagrad, which dynamically adjust
4529          learning rates and enhance convergence performance in diverse
4530          federated settings."
4531  },
4532  {
4533      "algorithm": "Elastic Aggregation",
4534      "description": "Classical federated optimization scheme that
4535          introduces elasticity in the aggregation process by assigning
4536          dynamic weights to client updates based on the sensitivity
4537          of model parameters. This balances stability and adaptability
4538          , improving performance on heterogeneous datasets."
4539  },
4540  {
4541      "algorithm": "FedFed",
4542      "description": "Federated learning algorithms that allows
4543          partial feature sharing between clients and server and

```

```

4536     mitigates data heterogeneity by distinguishing between
4537     performance-sensitive and performance-robust features and
4538     selectively distilling the former. This allows clients to
4539     retain useful features while benefiting from cross-client
4540     generalization."
4541   },
4542   {
4543     "algorithm": "pFedSim",
4544     "description": "Personalized Federated Learning Algorithm that
4545       enhances personalization by aggregating client models based
4546       on the similarity of their data distributions. Clients with
4547       more similar data contribute more significantly to each other
4548       's updates, enabling customized learning without explicit
4549       data sharing."
4550   },
4551   {
4552     "algorithm": "FedMD",
4553     "description": "Personalized Federated Learning Algorithm that
4554       supports clients with heterogeneous architectures by
4555       performing knowledge distillation using a shared public
4556       dataset. Clients align on output predictions rather than
4557       model parameters, enabling collaborative training without
4558       requiring architectural uniformity."
4559   },
4560   {
4561     "algorithm": "APFL",
4562     "description": "Personalized Federated Learning Algorithm that
4563       implements an adaptive mixing strategy where each client
4564       maintains both a local and a global model. The final model
4565       output is a weighted combination, and the mixing coefficient
4566       is learned during training to achieve optimal personalization
4567       ."
4568   },
4569   {
4570     "algorithm": "LG-FedAvg",
4571     "description": "Personalized Federated Learning Algorithm that
4572       decomposes models into local and global components, where
4573       only the global part is aggregated across clients. This
4574       preserves local knowledge while benefiting from global trends
4575       , supporting personalized learning in non-IID settings."
4576   },
4577   {
4578     "algorithm": "FedBN",
4579     "description": "Personalized Federated Learning Algorithm that
4580       keeps batch normalization layers local to each client while
4581       sharing the rest of the model globally. This enables
4582       adaptation to client-specific feature distributions and
4583       enhances performance under feature heterogeneity."
4584   },
4585   {
4586     "algorithm": "FedPer",
4587     "description": "Personalized Federated Learning Algorithm that
4588       introduces personalization by partitioning the model into a
4589       globally shared base and a locally updated head. This
       structure allows clients to fine-tune their models based on
       local data while retaining shared representations."
4590   },
4591   {
4592     "algorithm": "FedRep",
4593     "description": "Personalized Federated Learning Algorithm that
4594       learns a common feature extractor shared across clients and
4595       allows each client to train its own classifier head. This
4596       separation supports personalization without requiring full
4597       model updates across the federation."
4598   },
4599

```

```

4590
4591
4592
4593
4594
4595
4596
4597
4598
4599
4600
4601
4602
4603
4604
4605
4606
4607
4608
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
{
  "algorithm": "Per-FedAvg",
  "description": "Personalized Federated Learning Algorithm that combines meta-learning (specifically MAML) with federated learning to learn a global initialization that can be rapidly personalized to each clients local data, enabling quick adaptation with limited samples."
},
{
  "algorithm": "pFedMe",
  "description": "Personalized Federated Learning Algorithm that formulates personalized federated learning as a bi-level optimization problem using Moreau envelopes, which allows decoupling global and local updates. This improves convergence and supports better personalization."
},
{
  "algorithm": "FedEM",
  "description": "Personalized Federated Learning Algorithm that performs multi-task learning. It treats each clients model as part of a mixture of distributions and trains them via the Expectation-Maximization algorithm. This enables multi-task personalization by modeling shared and unique components across clients."
},
{
  "algorithm": "Ditto",
  "description": "Personalized Federated Learning Algorithm that simultaneously trains a global model for generalization and a personalized model for each client, ensuring fairness and robustness through dual-objective optimization."
},
{
  "algorithm": "pFedHN",
  "description": "Personalized Federated Learning Algorithm that utilizes a central hypernetwork that generates personalized model weights for clients, enabling parameter sharing while allowing client-specific adaptations."
},
{
  "algorithm": "pFedLA",
  "description": "Personalized Federated Learning Algorithm that performs layer-wise model aggregation, assigning personalized importance to each layer across clients to improve fine-grained adaptation in non-IID environments."
},
{
  "algorithm": "CFL",
  "description": "Federated Learning algorithm that clusters clients based on model or data similarity and trains distinct models per cluster to effectively manage heterogeneity across groups."
},
{
  "algorithm": "FedFomo",
  "description": "Personalized Federated Learning Algorithm that maintains a personalized model by aggregating updates from peer clients weighted by similarity scores, using a first-order gradient approximation to ensure communication efficiency."
},
{
  "algorithm": "FedBabu",
  "description": "Personalized Federated Learning Algorithm that improves personalized learning by aggregating only the shared

```

```

4644         body (feature extractor) of the model while keeping client-
4645         specific heads independent."
4646     },
4647     {
4648         "algorithm": "FedAP",
4649         "description": "Personalized Federated Learning Algorithm that
4650             employs adaptive batch normalization to tailor models to
4651             healthcare clients, effectively handling distribution shifts
4652             across medical institutions."
4653     },
4654     {
4655         "algorithm": "MetaFed",
4656         "description": "Personalized Federated Learning Algorithm that
4657             applies a cyclic knowledge distillation framework across
4658             federated groups, improving model generalizability without
4659             raw data exchange and without necessity of a server."
4660     },
4661     {
4662         "algorithm": "FedRoD",
4663         "description": "Regularization-based Federated Learning
4664             approach that balances the benefits of generalization and
4665             personalization by adaptively mixing global and local model
4666             components using regularized dual objectives."
4667     },
4668     {
4669         "algorithm": "FedProto",
4670         "description": "Personalized and generalizable Federated
4671             learning algorithm that aligns client features through the
4672             use of global class prototypes, promoting semantic
4673             consistency while preserving personalization."
4674     },
4675     {
4676         "algorithm": "FedALA",
4677         "description": "Personalized Federated learning algorithm that
4678             aggregates local models adaptively by learning relevance-
4679             based weights for each client, enabling better
4680             personalization through dynamic influence modeling."
4681     },
4682     {
4683         "algorithm": "PeFLL",
4684         "description": "Personalized Federated learning algorithm that
4685             incorporates meta-learning to personalize model updates for
4686             each client by learning an optimal initialization that
4687             generalizes quickly to local tasks."
4688     },
4689     {
4690         "algorithm": "FLUTE",
4691         "description": "Personalized Federated learning algorithm that
4692             addresses model underparameterization in resource-constrained
4693             environments by learning efficient global and local decoders
4694             for distributed representation learning."
4695     },
4696     {
4697         "algorithm": "FedAS",
4698         "description": "Personalized Federated learning algorithm using
4699             regularization-based approach that aligns global and local
4700             model updates using adaptive strategies to reduce
4701             inconsistency and improve convergence in personalized
4702             federated learning."
4703     },
4704     {
4705         "algorithm": "Floco",
4706         "description": "Personalized Federated learning algorithm that
4707             models client relationships using a graph of local modes and

```

```

4698     clusters them for collaborative training, leveraging shared
4699     structure without central data."
4700   },
4701   {
4702     "algorithm": "FedSR",
4703     "\"description\"": "Federated domain generalization-based technique
4704       that applies simple regularization across domain
4705       representations to improve out-of-distribution generalization
4706       in federated settings."
4707   },
4708   {
4709     "algorithm": "ADCOL",
4710     "\"description\"": "Federated domain generalization-based technique
4711       that uses adversarial learning to align feature spaces
4712       across clients, enabling domain generalization under non-IID
4713       conditions."
4714   },
4715   {
4716     "algorithm": "FedIIR",
4717     "\"description\"": "Federated domain generalization-based technique
4718       that identifies and leverages invariant relationships across
4719       domains to enhance generalization to out-of-distribution
4720       data in federated settings."
4721 }

```

4721 C.5 LLMS AS THE AGENT CORE COMPONENTS

4722 MODEL SELECTION JUSTIFICATION

4723 To assess the reasoning, planning, and tool-use capabilities of large language model (LLM) agents
 4724 in the context of real-world federated learning workflows, we evaluate a set of 24 LLMs on the
 4725 FedAgentBench suite. The selected models span both proprietary and open-source categories,
 4726 ensuring broad coverage across scale, training data diversity, and model access paradigms.

4727 We include 10 proprietary LLMs from leading industrial labs such as OpenAI and Anthropic,
 4728 including multiple variants of GPT-4. These models represent the current frontier of general-purpose
 4729 foundation models, often topping benchmarks in instruction-following, tool use, and reasoning. Their
 4730 inclusion allows us to benchmark state-of-the-art commercial performance in the agentic FL setting.

4731 We particularly include a range of GPT-family models developed by OpenAI to cover both ends of
 4732 the performance-efficiency spectrum in proprietary large language models (LLMs). The rationale is
 4733 threefold:

4734 **(i) Proven Instruction-Following and Reasoning Abilities:**

4735 GPT-4 and its variants have consistently demonstrated state-of-the-art performance across multiple
 4736 benchmarks involving instruction following, task decomposition, and multi-step reasoning capabilities
 4737 essential for evaluating LLM agents in complex federated learning pipelines such as FedAgentBench.

4738 **(ii) Variants across Performance Tiers and Costs:**

4739 The selection spans high-end models (e.g., GPT-4.1, GPT-4o) and lightweight alternatives (e.g.,
 4740 GPT-4.1-mini, GPT-o3-mini). This allows us to study the trade-offs between agent reasoning quality
 4741 and computational/resource efficiency, particularly relevant for real-world FL deployment where cost
 4742 and inference speed matter.

4743 **(iii) Industry Adoption and API Availability:**

4744 These models are widely adopted in both academic and industrial applications and offer stable,
 4745 reproducible APIs. This ensures consistent evaluation and compatibility with tool-augmented LLM
 4746 agent frameworks.

4747 Besides, we evaluate 14 open-source LLMs across four major families: LLaMA, DeepSeek, Qwen,
 4748 and Gemma. These models are chosen for their state-of-the-art performance in open benchmarks,
 4749 availability in multiple parameter scales (from 9B to 685B), and varying architectural innovations

(e.g., distillation in DeepSeek, instruction tuning in Qwen, and scalability in Gemma). This selection ensures a representative spectrum of recent advances in open-source LLM development, and provides insight into how scale, family, and fine-tuning affect FL-agent performance.

By including both proprietary and open models across diverse sizes and pretraining paradigms, our evaluation is designed to offer fair, scalable, and realistic comparisons, while informing the community of strengths and limitations across model categories in complex multi-agent settings like FedAgentBench.

Table 8: Descriptions for Proprietary LLMs in FedAgentBench

Model	Description	Capabilities	Use Rationale	Caveats / Notes
GPT-4.1	Latest high-performance model from OpenAI with advanced reasoning and planning.	Chain-of-thought reasoning, tool use, structured outputs.	Reference proprietary agent for end-to-end workflows.	High cost and latency; not ideal for real-time execution.
GPT-4o	Multimodal flagship model supporting vision-language tasks.	Multilingual, tool calling, multimodal reasoning.	Evaluated for vision + tool scenarios.	New model; some outputs may vary between calls.
GPT-4	Original GPT-4 model with top-tier generalization.	Long-context, reasoning, structured outputs.	Used as baseline for reasoning accuracy.	Slower than turbo and newer variants.
GPT-4-Turbo	Faster and cheaper version of GPT-4 for API use.	Efficient inference, similar capabilities to GPT-4.	Preferred when cost is a concern.	Slightly less coherent outputs.
GPT-4.1-mini	Distilled variant optimized for fast inference.	Good single-step logic, mid-range planning.	Used in real-time assistant agents.	Weaker on edge-case and ambiguous tasks.
GPT-4o-mini	Smaller variant of GPT-4o with multimodal support.	Vision-language support, low-latency.	Benchmarked in low-resource multimodal agents.	Reduced performance in logic-intensive tasks.
GPT-o4-mini	Lightweight GPT-4 style model.	Text generation and simple instructions.	Ablation studies for low-cost GPT agents.	Unclear origin; may alias other mini variants.
GPT-o3-mini	GPT-3.5-based efficient variant.	Very fast, single-turn chat.	Used for comparison with older architectures.	Weak reasoning; not reliable for planning.
GPT-3.5 Turbo	Predecessor to GPT-4, cheaper and widely used.	Fast, capable for basic instruction and QA.	Low-cost reference for proprietary agents.	Token alignment issues in structured tasks.
Claude-3.7 Sonnet	Mid-size model from Anthropic with alignment tuning.	Safety-aligned generation, multilingual, tool use.	Benchmarked against non-OpenAI proprietary model.	Slightly lower fluency than top Claude variants.

D RESULTS AND DISCUSSIONS

We conducted extensive evaluations of both proprietary and open-source LLM agents across 6 environments, out of which the success rates for Histopathology have been mentioned in the main paper. The success rates for the remaining 5 environments *viz.*, Dermatology, Ultrasound, MRI, Fundus and X-Ray environments are reported here. The results of these experiments are presented in Tables 10-15. These tables capture performance under two paradigms: fine-grained multi-step guidance and goal-oriented single-shot instruction, revealing consistent trends across modalities. Notably, the independent script generation setting in Table 12 illustrates a sharp decline in performance for most agents, underscoring the challenges of long-horizon task planning without explicit decomposition. Overall Time-requirement metrics for task resolution are summarized in Table 16, providing a holistic view of capability and practicality across LLM variants. Figs 10-36 show snippets of different phases of the FL workflow with various LLMs and different imaging modalities which help to understand their success and failure modes.

D.1 DISCUSSION ON AGENTIC PERFORMANCE IN INDIVIDUAL HEALTHCARE ENVIRONMENT

The overall comparative agentic performance in all environments has been summarized in Table 17. Furthermore, we also analyze the performance of individual environments. Table 10 reports the performance of open-source and proprietary LLM agents in the **Dermatology environment**. Proprietary models obtain the strongest results under both guidance regimes. **GPT-4.1** is the highest-performing system, achieving consistent 5/5 scores on most sub-tasks and the highest **Overall** performance (**94.29** with fine-grained guidance; **88.57** with goal-oriented guidance).

4806
 4807
 4808
 4809
 4810
 4811
 4812
 4813
 4814
 4815
 4816
 4817
 4818
 4819

Table 9: Descriptions for Open-Source LLMs in FedAgentBench

Model	Description	Capabilities	Use Rationale	Caveats / Notes
LLaMA-4 Maverick	Latest LLaMA release (2025) with top-tier accuracy in reasoning and instruction following.	Instruction following, long-context reasoning, coding tasks.	Used for evaluating high-end open-source agents.	Resource heavy; slower than lighter LLaMA variants.
LLaMA-4 Scout	2025 LLaMA-4 variant optimized for cost-efficient inference.	Balanced reasoning and fast response for system agents.	Used as mid-range open-source agent in system and logic tasks.	Less expressive than Maverick.
LLaMA-3 70B	Flagship LLaMA model (2024) with extensive instruction tuning.	Reasoning, multilingual tasks, tool use.	Used for top-tier open-source evaluation.	Less performant than newer LLaMA-4 variants.
LLaMA-3 8B	Smaller variant of LLaMA-3 for constrained environments.	General understanding, good for fast responses.	Used in real-time benchmarking of lighter agents.	Limited capacity in multi-hop reasoning.
DeepSeek-V3	Latest release from DeepSeek with strong Chinese-English capability.	Multilingual chat, code, reasoning.	Used to test multilingual and cross-domain agents.	Less stable tool usage.
DeepSeek-R1	General purpose 2024 DeepSeek model.	Basic LLM tasks, reasoning.	Baseline open-source reference.	Lower precision under stress tests.
DeepSeek-R1-Distill	Distilled version of DeepSeek-R1 on LLaMA-70B.	Fast inference, low-resource usage.	Used in lightweight evaluations.	Lower performance ceiling.
Qwen 3 235B	Massive MoE model by Alibaba; high capacity and strong multilingual.	Multilingual, few-shot generalization, long context.	Benchmarked as high-capacity open-source agent.	Costly to run, sparse documentation.
Qwen QwQ 32B	Intermediate-sized multilingual Qwen model.	Instruction following, QA, multilingual chat.	Used as cost-performance mid-range Qwen agent.	Less stable tool usage.
Qwen 3 30B	Well-balanced Qwen variant.	Reliable output, structured reasoning.	Used in systems requiring stable decoding.	Reduced multilingual coverage vs 235B.
Qwen 3 14B	Smaller Qwen for lightweight use.	Quick single-turn tasks.	Used in sub-agents and pre-filtering roles.	Shallow reasoning, poor long-context.
Gemma 3 27B Instruct	Instruction-tuned model by Google.	Tool use, summarization, chat.	Tested for logic tasks.	Less capable in multi-modal domains.
Gemma 3 12B Instruct	Smaller Gemma variant.	Common NLP tasks.	System-level fast agent.	May misfire structured outputs.
Gemma 2 9B Instruct	Previous generation Gemma model.	Lightweight inference.	Tested in low-cost agent scenarios.	Lowest instruction accuracy among Geminis.

4848
 4849
 4850
 4851
 4852
 4853
 4854
 4855
 4856
 4857
 4858
 4859

4860 Table 10: Comparison of open-source and proprietary LLM agents across different stages of federated
4861 learning: Client Selection (Client-Sel), Data Pre-processing (Data-Pre), Label Harmonization (Label-
4862 Harm), and Federated Training (Fed-Train) in **Dermatology** environment based on skin cancer
4863 detection task. a/b refers to the proportion of successful runs 'a' out of the total number of runs 'b'

4864	4865	4866	Fine-grained guidance					Goal-oriented guidance				
			Client-Sel	Data-Pre	Label-Harm	Fed-Train	Overall	Client-Sel	Data-Pre	Label-Harm	Fed-Train	Overall
			a_1, a_2, a_3	a_4	a_5	a_6, a_7		a_1, a_2, a_3	a_4	a_5	a_6, a_7	
Proprietary Models												
GPT-4.1	5/5, 5/5, 5/5	5/5	3/5	5/5, 5/5	94.29		5/5, 4/5, 5/5	5/5	3/5	4/5, 5/5	88.57	
GPT-4o	5/5, 3/5, 5/5	5/5	1/5	1/5, 5/5	71.43		5/5, 1/5, 5/5	5/5	1/5	1/5, 5/5	65.71	
GPT-4	5/5, 4/5, 5/5	0/5	1/5	3/5, 5/5	65.71		5/5, 1/5, 5/5	0/5	0/5	2/5, 5/5	51.43	
GPT-4-Turbo	5/5, 3/5, 5/5	2/5	1/5	3/5, 5/5	68.57		5/5, 3/5, 5/5	5/5	1/5	2/5, 5/5	74.29	
GPT-4.1-mini	5/5, 5/5, 5/5	5/5	3/5	3/5, 5/5	88.57		5/5, 5/5, 5/5	3/5	3/5	3/5, 5/5	82.86	
GPT-4o-mini	5/5, 1/5, 3/5	5/5	3/5	3/5, 4/5	68.57		5/5, 0/5, 3/5	5/5	1/5	2/5, 4/5	57.14	
GPT-4o-mini	5/5, 4/5, 5/5	5/5	3/5	3/5, 5/5	85.71		5/5, 3/5, 5/5	4/5	2/5	3/5, 4/5	74.29	
GPT-o3-mini	5/5, 3/5, 5/5	0/5	2/5	3/5, 5/5	65.71		5/5, 1/5, 5/5	0/5	2/5	3/5, 5/5	60.00	
GPT-3.5-Turbo	5/5, 0/5, 0/5	0/5	0/5	1/5, 3/5	25.71		5/5, 0/5, 0/5	2/5	0/5	1/5, 3/5	31.43	
Claude-3-7-Sonnet	5/5, 2/5, 3/5	2/5	1/5	2/5, 3/5	51.42		5/5, 2/5, 3/5	2/5	1/5	2/5, 5/5	57.14	
Open-source Models												
Huge Models												
DeepSeek-V3	5/5, 1/5, 5/5	5/5	5/5	4/5, 5/5	85.71		5/5, 1/5, 5/5	4/5	4/5	4/5, 5/5	80.00	
DeepSeek-R1	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86		5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.85	
Qwen3 235B	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86		5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.85	
LLaMA-4 Maverick	5/5, 1/5, 4/5	5/5	3/5	2/5, 5/5	71.43		5/5, 1/5, 4/5	5/5	3/5, 5/5	74.29		
LLaMA-4 Scout	5/5, 1/5, 5/5	5/5	3/5	2/5, 5/5	74.29		5/5, 2/5, 5/5	5/5	3/5	2/5, 5/5	77.14	
Large Models												
DeepSeek-R1-70B	5/5, 0/5, 5/5	0/5	0/5	1/5, 5/5	45.71		5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86	
LLaMA-3-70B	5/5, 0/5, 5/5	1/5	1/5	2/5, 5/5	54.29		5/5, 0/5, 5/5	2/5	2/5	1/5, 5/5	57.14	
Medium Models												
Qwen QwQ 32B	5/5, 4/5, 5/5	5/5	4/5	4/5, 5/5	91.43		5/5, 4/5, 5/5	5/5	3/5	3/5, 5/5	85.71	
Qwen3-30B	5/5, 0/5, 5/5	0/5	0/5	1/5, 5/5	45.71		5/5, 0/5, 5/5	0/5	0/5	1/5, 5/5	45.71	
Gemma3-27B-instruct	5/5, 0/5, 0/5	0/5	0/5	0/5, 0/5	14.29		5/5, 0/5, 0/5	0/5	0/5	0/5, 0/5	14.29	
Small Models												
Gemma-2-9B	5/5, 0/5, 5/5	1/5	1/5	1/5, 5/5	51.43		5/5, 0/5, 5/5	1/5	1/5	1/5, 5/5	51.43	
LLaMA-3-8B	5/5, 0/5, 5/5	5/5	2/5	1/5, 5/5	65.71		5/5, 0/5, 5/5	5/5	2/5	1/5, 5/5	65.71	
Qwen3-14B	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86		5/5, 0/5, 5/5	0/5	0/5	0/5, 4/5	40.00	
Gemma3-12B-instruct	5/5, 0/5, 0/5	0/5	0/5	0/5, 0/5	14.29		5/5, 0/5, 0/5	0/5	0/5	0/5, 0/5	14.29	

4886 Table 11: Comparison of open-source and Proprietary LLM agents in **Ultrasound** environment for
4887 **breast cancer detection task**

4889	4890	4891	Fine-grained guidance					Goal-oriented guidance				
			Client-Sel	Data-Pre	Label-Harm	Fed-Train	Overall	Client-Sel	Data-Pre	Label-Harm	Fed-Train	Overall
			S_1, C_1, S_2	C_2	C_3	S_3, S_4		S_1, C_1, S_2	C_2	C_3	S_3, S_4	
Proprietary Models												
GPT-4.1	5/5, 3/5, 5/5	5/5	5/5	5/5, 5/5	94.29		5/5, 3/5, 5/5	5/5	5/5	5/5, 5/5	94.29	
GPT-4o	5/5, 0/5, 5/5	5/5	3/5	1/5, 5/5	68.57		5/5, 0/5, 5/5	5/5	2/5	1/5, 5/5	65.71	
GPT-4	5/5, 3/5, 5/5	1/5	1/5	3/5, 5/5	65.71		5/5, 3/5, 5/5	0/5	1/5	3/5, 5/5	62.86	
GPT-4-Turbo	5/5, 3/5, 5/5	1/5	1/5	3/5, 5/5	65.71		5/5, 3/5, 5/5	4/5	1/5	3/5, 5/5	74.29	
GPT-4.1-mini	5/5, 3/5, 5/5	5/5	3/5	4/5, 5/5	85.71		5/5, 2/5, 5/5	3/5	4/5	3/5, 5/5	77.14	
GPT-4o-mini	5/5, 1/5, 3/5	5/5	3/5	3/5, 5/5	71.43		5/5, 1/5, 3/5	5/5	1/5	5/5, 5/5	71.43	
GPT-4o-mini	5/5, 3/5, 5/5	5/5	3/5	4/5, 5/5	85.71		5/5, 3/5, 5/5	4/5	3/5	4/5, 5/5	82.86	
GPT-03-mini	5/5, 2/5, 5/5	1/5	1/5	3/5, 4/5	60.00		5/5, 1/5, 5/5	1/5	2/5	3/5, 5/5	62.86	
GPT-3.5-Turbo	5/5, 0/5, 0/5	0/5	0/5	1/5, 3/5	25.71		5/5, 0/5, 0/5	2/5	0/5	1/5, 4/5	34.29	
Claude-3-7	5/5, 2/5, 3/5	2/5	1/5	3/5, 3/5	54.29		5/5, 2/5, 3/5	2/5	1/5	3/5, 3/5	54.29	
Open-source Models												
Huge Models												
DeepSeek-V3	5/5, 3/5, 5/5	5/5	5/5	4/5, 5/5	91.43		5/5, 2/5, 5/5	4/5	5/5	4/5, 5/5	85.71	
DeepSeek-R1	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86		5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86	
Qwen3 235B	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86		5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86	
LLaMA-4 Maverick	5/5, 3/5, 5/5	5/5	4/5	4/5, 5/5	88.57		5/5, 3/5, 5/5	5/5	3/5	3/5, 5/5	82.86	
LLaMA-4 Scout	5/5, 3/5, 5/5	5/5	4/5	3/5, 5/5	85.71		5/5, 1/5, 5/5	5/5	3/5	2/5, 5/5	74.28	
Large Models												
DeepSeek-R1-70B	5/5, 3/5, 5/5	3/5	1/5	2/5, 5/5	74.28		5/5, 1/5, 5/5	3/5	0/5	2/5, 5/5	68.57	
LLaMA-3-70B	5/5, 3/5, 5/5	2/5	2/5	2/5, 5/5	68.57		5/5, 3/5, 5/5	2/5	2/5	2/5, 5/5	68.57	
Medium Models												
Qwen QwQ 32B	5/5, 3/5, 5/5	4/5	4/5	4/5, 5/5	85.71		5/5, 3/5, 5/5	2/5	4/5	4/5, 5/5	80.00	
Qwen3-30B	5/5, 0/5, 5/5	0/5	0/5	1/5, 5/5	45.71		5/5, 0/5, 5/5	0/5	0/5	1/5, 5/5	45.71	
Gemma3-27B-instruct	5/5, 0/5, 0/5	0/5	0/5	0/5, 0/5	14.29		5/5, 0/5, 0/5	0/5	0/5	0/5, 0/5	14.29	
Small Models												
Gemma-2-9B	5/5, 0/5, 4/5	2/5	1/5	1/5, 5/5	51.43		5/5, 0/5, 4/5	1/5	1/5	1/5, 5/5	48.57	
LLaMA-3-8B	5/5, 0/5, 4/5	4/5	2/5	1/5, 5/5	60.00		5/5, 0/5, 4/5	4/5	2/5	1/5, 5/5	60.00	
Qwen3-14B	5/5, 0/5, 0/5	0/5	0/5	0/5, 5/5	28.57		5/5, 0/5, 0/5	0/5	0/5	0/5, 4/5	25.71	
Gemma3-12B-instruct	5/5, 0/5, 0/5	0/5	0/5	0/5, 0/5	14.29		5/5, 0/5, 0/5	0/5	0/5	0/5, 0/5	14.29	

4912 A second performance tier includes **GPT-4.1-mini**, **GPT-04-mini**, **GPT-4o**, **GPT-4-Turbo**, **GPT-03-mini**, and **GPT-4**. **GPT-3.5-Turbo** shows substantially lower accuracy, and **Claude-3-7-Sonnet**
4913 ranks in the middle range.

Table 12: Comparison of open-source and Proprietary LLM agents for **breast cancer detection task** in **Ultrasound** environment on independent script generation for solving individual task.

Model	Client-Sel	Data-Pre	Label-Harm	Fed-Train	Overall
Proprietary Models					
GPT-4.1	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86
GPT-4o	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86
GPT-4	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86
GPT-4-Turbo	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86
GPT-4.1-mini	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86
GPT-4o-mini	5/5, 0/5, 3/5	0/5	0/5	0/5, 5/5	37.14
GPT-o4-mini	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86
GPT-03-mini	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86
GPT-3.5-Turbo	5/5, 0/5, 0/5	0/5	0/5	0/5, 4/5	25.71
Claude-3-7	5/5, 0/5, 3/5	0/5	0/5	0/5, 3/5	31.43
Open-source Models					
Huge Models					
DeepSeek-V3	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86
DeepSeek-R1	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86
Qwen3 235B	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86
LLaMA-4 Maverick	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86
LLaMA-4 Scout	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86
Large Models					
DeepSeek-R1-70B	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86
LLaMA-3-70B	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86
Medium Models					
Qwen QwQ 32B	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86
Qwen3-30B	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86
Gemma3-27B-instruct	5/5, 0/5, 0/5	0/5	0/5	0/5, 0/5	14.29
Small Models					
Gemma-2-9B	5/5, 0/5, 4/5	0/5	0/5	0/5, 5/5	40.0
LLaMA-3-8B	5/5, 0/5, 4/5	0/5	0/5	0/5, 5/5	40.0
Qwen-3-14B	5/5, 0/5, 0/5	0/5	0/5	0/5, 4/5	25.71
Gemma3-12B-instruct	5/5, 0/5, 0/5	0/5	0/5	0/5, 0/5	14.29

Table 13: Comparison of open-source and Proprietary LLM agents for **brain tumor detection task** in **MRI** environment

Model	Fine-grained guidance					Goal-oriented guidance				
	Client-Sel	Data-Pre	Label-Harm	Fed-Train	Overall	Client-Sel	Data-Pre	Label-Harm	Fed-Train	Overall
GPT-4.1	5/5, 5/5, 5/5	5/5	5/5	5/5, 5/5	100.00	5/5, 5/5, 5/5	5/5	5/5	5/5, 5/5	100.00
GPT-4o	5/5, 3/5, 5/5	5/5	4/5	1/5, 5/5	71.43	5/5, 3/5, 5/5	5/5	3/5	1/5, 5/5	68.57
GPT-4	5/5, 5/5, 5/5	1/5	2/5	3/5, 5/5	71.43	5/5, 4/5, 5/5	0/5	1/5	3/5, 5/5	65.71
GPT-4-Turbo	5/5, 5/5, 5/5	1/5	2/5	3/5, 5/5	71.43	5/5, 4/5, 5/5	4/5	1/5	3/5, 5/5	77.14
GPT-4.1-mini	5/5, 4/5, 5/5	5/5	3/5	4/5, 5/5	88.57	5/5, 3/5, 5/5	3/5	3/5	3/5, 5/5	77.14
GPT-4o-mini	5/5, 3/5, 3/5	5/5	3/5	3/5, 5/5	77.14	5/5, 2/5, 3/5	5/5	2/5	5/5, 5/5	74.29
GPT-04-mini	5/5, 5/5, 5/5	5/5	3/5	4/5, 5/5	91.43	5/5, 4/5, 5/5	4/5	2/5	4/5, 5/5	85.71
GPT-03-mini	5/5, 5/5, 5/5	1/5	1/5	4/5, 4/5	71.42	5/5, 4/5, 5/5	1/5	1/5	4/5, 5/5	74.29
GPT-3.5-Turbo	5/5, 0/5, 0/5	0/5	1/5	1/5, 3/5	25.71	5/5, 0/5, 0/5	2/5	0/5	1/5, 4/5	34.29
Claude-3-7	5/5, 4/5, 3/5	2/5	1/5	4/5, 3/5	57.14	5/5, 3/5, 3/5	2/5	1/5	3/5, 3/5	57.14
Open-source Models										
Huge Models										
DeepSeek-V3	5/5, 4/5, 5/5	5/5	5/5	4/5, 5/5	94.29	5/5, 3/5, 5/5	4/5	5/5	4/5, 5/5	88.57
DeepSeek-R1	5/5, 2/5, 5/5	0/5	0/5	0/5, 5/5	42.86	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86
Qwen3 235B	5/5, 2/5, 5/5	0/5	0/5	0/5, 5/5	42.86	5/5, 1/5, 5/5	0/5	0/5	0/5, 5/5	42.86
LLaMA-4 Maverick	5/5, 5/5, 5/5	5/5	4/5	4/5, 5/5	94.29	5/5, 4/5, 5/5	5/5	3/5	3/5, 5/5	85.71
LLaMA-4 Scout	5/5, 4/5, 5/5	5/5	4/5	2/5, 5/5	85.71	5/5, 3/5, 5/5	5/5	3/5	2/5, 5/5	74.29
Large Models										
DeepSeek-R1-70B	5/5, 5/5, 5/5	3/5	1/5	2/5, 5/5	74.29	5/5, 4/5, 5/5	3/5	0/5	2/5, 5/5	68.57
LLaMA-3-70B	5/5, 4/5, 5/5	2/5	2/5	2/5, 5/5	71.43	5/5, 4/5, 5/5	2/5	2/5	2/5, 5/5	71.43
Medium Models										
Qwen QwQ 32B	5/5, 4/5, 5/5	4/5	4/5	4/5, 5/5	88.57	5/5, 4/5, 5/5	2/5	4/5	4/5, 5/5	82.86
Qwen3-30B	5/5, 2/5, 5/5	0/5	0/5	1/5, 5/5	48.57	5/5, 0/5, 5/5	0/5	0/5	1/5, 5/5	45.71
Gemma3-27B-instruct	5/5, 1/5, 2/5	0/5	0/5	0/5, 0/5	14.29	5/5, 0/5, 2/5	0/5	0/5	0/5, 0/5	14.29
Small Models										
Gemma-2-9B	5/5, 1/5, 4/5	2/5	1/5	1/5, 5/5	51.43	5/5, 1/5, 4/5	1/5	1/5	1/5, 5/5	48.57
LLaMA-3-8B	5/5, 3/5, 4/5	4/5	2/5	1/5, 5/5	62.86	5/5, 2/5, 4/5	4/5	2/5	1/5, 5/5	60.00
Qwen-3-14B	5/5, 1/5, 2/5	0/5	0/5	0/5, 5/5	28.57	5/5, 0/5, 2/5	0/5	0/5	0/5, 4/5	25.71
Gemma3-12B-instruct	5/5, 1/5, 2/5	0/5	0/5	0/5, 0/5	14.29	5/5, 0/5, 2/5	0/5	0/5	0/5, 0/5	14.29

Across stages, higher-capacity models are most reliable on **Client Selection** and **Federated Training**, frequently achieving perfect scores (5/5, 5/5). Performance degrades most notably on **Data Pre-processing** and especially **Label Harmonization**, where mid-tier and smaller models often obtain 0/5 or 1/5, reducing their **Overall** scores even when later stages are solved correctly.

Among open-source systems, **DeepSeek-V3** performs best (85.71 / 80.00) with comparatively balanced behavior across stages. **LLaMA-4 Maverick** and **LLaMA-4 Scout** form the next group (71.43–85.71 depending on guidance). Lower-capacity or less-aligned open-source models (e.g., **DeepSeek-R1**, **Qwen3-235B**, **Gemma3-12B-instruct**) frequently fail in early pipeline stages and therefore yield the lowest scores.

Table 14: Comparison of open-source and Proprietary LLM agents for **Glaucoma detection task** in **Fundus** environment

Model	Fine-grained guidance					Goal-oriented guidance						
	Client-Sel		Data-Pre	Label-Harm	Fed-Train	Overall	Client-Sel		Data-Pre	Label-Harm	Fed-Train	Overall
	a_1, a_2, a_3		a_4	a_5	a_6, a_7		a_1, a_2, a_3		a_4	a_5	a_6, a_7	
Proprietary Models												
GPT-4.1	5/5, 5/5, 5/5	5/5	5/5	4/5, 5/5	97.14	5/5, 4/5, 5/5	5/5	5/5	4/5, 5/5	94.29		
GPT-4o	5/5, 2/5, 5/5	5/5	3/5	1/5, 5/5	74.29	5/5, 2/5, 5/5	5/5	3/5	1/5, 5/5	74.29		
GPT-4	5/5, 4/5, 5/5	1/5	1/5	3/5, 5/5	68.57	5/5, 4/5, 5/5	0/5	1/5	3/5, 5/5	65.71		
GPT-4-Turbo	5/5, 4/5, 5/5	1/5	1/5	3/5, 5/5	68.57	5/5, 4/5, 5/5	4/5	1/5	3/5, 5/5	77.14		
GPT-4.1-mini	5/5, 4/5, 5/5	5/5	3/5	4/5, 5/5	88.57	5/5, 2/5, 5/5	3/5	4/5	3/5, 5/5	77.14		
GPT-4o-mini	5/5, 3/5, 3/5	5/5	3/5	3/5, 5/5	77.14	5/5, 2/5, 3/5	5/5	1/5	4/5, 5/5	71.43		
GPT-04-mini	5/5, 4/5, 5/5	5/5	3/5	4/5, 5/5	88.57	5/5, 4/5, 5/5	4/5	3/5	4/5, 5/5	85.71		
GPT-03-mini	5/5, 4/5, 5/5	1/5	1/5	4/5, 4/5	68.57	5/5, 4/5, 5/5	1/5	2/5	4/5, 5/5	74.29		
GPT-3.5-Turbo	5/5, 0/5, 0/5	0/5	0/5	1/5, 3/5	25.71	5/5, 0/5, 0/5	2/5	0/5	1/5, 4/5	34.29		
Claude-3-7	5/5, 3/5, 3/5	2/5	1/5	3/5, 3/5	57.14	5/5, 3/5, 3/5	2/5	1/5	3/5, 3/5	57.14		
Open-source Models												
Huge Models												
DeepSeek-V3	5/5, 4/5, 5/5	5/5	5/5	4/5, 5/5	94.29	5/5, 3/5, 5/5	4/5	5/5	4/5, 5/5	88.57		
DeepSeek-R1	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86		
Qwen3 235B	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86		
LLaMA-4 Maverick	5/5, 4/5, 5/5	5/5	4/5	4/5, 5/5	91.43	5/5, 4/5, 5/5	5/5	3/5	3/5, 5/5	85.71		
LLaMA-4 Scout	5/5, 4/5, 5/5	5/5	4/5	2/5, 5/5	85.71	5/5, 1/5, 5/5	5/5	3/5	2/5, 5/5	74.28		
Large Models												
DeepSeek-R1-70B	5/5, 4/5, 5/5	3/5	1/5	2/5, 5/5	71.43	5/5, 4/5, 5/5	3/5	0/5	2/5, 5/5	68.57		
LLaMA-3-70B	5/5, 4/5, 5/5	2/5	2/5	2/5, 5/5	71.43	5/5, 4/5, 5/5	2/5	2/5	2/5, 5/5	71.43		
Medium Models												
Qwen QwQ 32B	5/5, 4/5, 5/5	4/5	4/5	4/5, 5/5	88.57	5/5, 4/5, 5/5	2/5	4/5	4/5, 5/5	82.86		
Qwen3-30B	5/5, 1/5, 5/5	0/5	0/5	1/5, 5/5	48.57	5/5, 0/5, 5/5	0/5	0/5	1/5, 5/5	45.71		
Gemma3-27B-instruct	5/5, 0/5, 0/5	0/5	0/5	0/5, 0/5	14.29	5/5, 0/5, 0/5	0/5	0/5	0/5, 0/5	14.29		
Small Models												
Gemma-2-9B	5/5, 0/5, 4/5	2/5	1/5	1/5, 5/5	51.43	5/5, 0/5, 4/5	1/5	1/5	1/5, 5/5	48.57		
LLaMA-3-8B	5/5, 1/5, 4/5	4/5	2/5	1/5, 5/5	62.86	5/5, 0/5, 4/5	4/5	2/5	1/5, 5/5	60.00		
Qwen3-14B	5/5, 0/5, 0/5	0/5	0/5	0/5, 5/5	28.57	5/5, 0/5, 0/5	0/5	0/5	0/5, 4/5	25.71		
Gemma3-12B-instruct	5/5, 0/5, 0/5	0/5	0/5	0/5, 0/5	14.29	5/5, 0/5, 0/5	0/5	0/5	0/5, 0/5	14.29		

Table 15: Comparison of open-source and Proprietary LLM agents for **pneumonia detection task** in **chest X-Ray** environment

Model	Fine-grained guidance					Goal-oriented guidance						
	Client-Sel		Data-Pre	Label-Harm	Fed-Train	Overall	Client-Sel		Data-Pre	Label-Harm	Fed-Train	Overall
	a_1, a_2, a_3		a_4	a_5	a_6, a_7		a_1, a_2, a_3		a_4	a_5	a_6, a_7	
Proprietary Models												
GPT-4.1	5/5, 5/5, 5/5	5/5	5/5	5/5, 5/5	100.00	5/5, 5/5, 5/5	5/5	5/5	5/5, 5/5	100.00		
GPT-4o	5/5, 1/5, 5/5	5/5	3/5	1/5, 5/5	71.43	5/5, 1/5, 5/5	5/5	2/5	1/5, 5/5	68.57		
GPT-4	5/5, 5/5, 5/5	1/5	1/5	3/5, 5/5	71.43	5/5, 4/5, 5/5	0/5	1/5	3/5, 5/5	65.71		
GPT-4-Turbo	5/5, 5/5, 5/5	1/5	1/5	3/5, 5/5	71.43	5/5, 4/5, 5/5	4/5	1/5	3/5, 5/5	77.14		
GPT-4.1-mini	5/5, 4/5, 5/5	5/5	3/5	4/5, 5/5	88.57	5/5, 2/5, 5/5	3/5	4/5	3/5, 5/5	77.14		
GPT-4o-mini	5/5, 3/5, 3/5	5/5	3/5	3/5, 5/5	77.14	5/5, 2/5, 3/5	5/5	1/5	5/5, 5/5	74.29		
GPT-04-mini	5/5, 5/5, 5/5	5/5	3/5	4/5, 5/5	91.43	5/5, 4/5, 5/5	4/5	3/5	4/5, 5/5	85.71		
GPT-03-mini	5/5, 5/5, 5/5	1/5	1/5	4/5, 4/5	71.42	5/5, 4/5, 5/5	1/5	2/5	4/5, 5/5	74.29		
GPT-3.5-Turbo	5/5, 0/5, 0/5	0/5	0/5	1/5, 3/5	25.71	5/5, 0/5, 0/5	2/5	0/5	1/5, 4/5	34.29		
Claude-3-7	5/5, 3/5, 3/5	2/5	1/5	3/5, 3/5	57.14	5/5, 3/3, 3/5	2/5	1/5	3/5, 3/5	57.14		
Open-source Models												
Huge Models												
DeepSeek-V3	5/5, 4/5, 5/5	5/5	5/5	4/5, 5/5	94.29	5/5, 3/5, 5/5	4/5	5/5	4/5, 5/5	88.57		
DeepSeek-R1	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86		
Qwen3 235B	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86	5/5, 0/5, 5/5	0/5	0/5	0/5, 5/5	42.86		
LLaMA-4 Maverick	5/5, 5/5, 5/5	5/5	4/5	4/5, 5/5	94.29	5/5, 4/5, 5/5	5/5	3/5	3/5, 5/5	85.71		
LLaMA-4 Scout	5/5, 4/5, 5/5	5/5	4/5	2/5, 5/5	85.71	5/5, 1/5, 5/5	5/5	3/5	2/5, 5/5	74.28		
Large Models												
DeepSeek-R1-70B	5/5, 5/5, 5/5	3/5	1/5	2/5, 5/5	74.28	5/5, 4/5, 5/5	3/5	0/5	2/5, 5/5	68.57		
LLaMA-3-70B	5/5, 4/5, 5/5	2/5	2/5	2/5, 5/5	71.43	5/5, 4/5, 5/5	2/5	2/5	2/5, 5/5	71.43		
Medium Models												
Qwen QwQ 32B	5/5, 4/5, 5/5	4/5	4/5	4/5, 5/5	88.57	5/5, 4/5, 5/5	2/5	4/5	4/5, 5/5	82.86		
Qwen3-30B	5/5, 1/5, 5/5	0/5	0/5	1/5, 5/5	48.57	5/5, 0/5, 5/5	0/5	0/5	1/5, 5/5	45.71		
Gemma3-27B-instruct	5/5, 0/5, 0/5	0/5	0/5	0/5, 0/5	14.29	5/5, 0/5, 0/5	0/5	0/5	0/5, 0/5	14.29		
Small Models												
Gemma-2-9B	5/5, 0/5, 4/5	2/5	1/5	1/5, 5/5	51.43	5/5, 0/5, 4/5	1/5	1/5	1/5, 5/5	48.57		
LLaMA-3-8B	5/5, 1/5, 4/5	4/5	2/5	1/5, 5/5	62.86	5/5, 0/5, 4/5	4/5	2/5	1/5, 5/5	60.00		
Qwen3-14B	5/5, 0/5, 0/5	0/5	0/5	0/5, 5/5	28.57	5/5, 0/5, 0/5	0/5	0/5	0/5, 4/5	25.71		
Gemma3-12B-instruct	5/5, 0/5, 0/5	0/5	0/5	0/5, 0/5	14.29	5/5, 0/5, 0/5	0/5	0/5	0/5, 0/5	14.29		

Finally, **fine-grained guidance** consistently improves **overall** performance compared to **goal-oriented guidance**, indicating that explicit stepwise instructions help agents navigate the multi-stage federated learning workflow in Dermatology more effectively.

Table 11 compares open-source and proprietary LLM agents in the **Ultrasound environment** for breast cancer detection task under two guidance paradigms: fine-grained guidance, where each subtask is explicitly defined and goal-oriented guidance, where the model is only given the overall objective. Each model's performance is evaluated on four core subtasks, and the final column

5022
 5023
 5024
 5025
 5026
 5027
 5028
 5029
 5030
 5031
 5032
 5033
 5034

5035 Table 16: Comparison of average time taken by each agent to solve respective tasks (in seconds)
 5036 using different LLMs.
 5037

Model	S_1	C_1	S_2	C_2	C_3	S_3	S_4
Proprietary Models							
GPT-4.1	1.8	64.8	55.5	302.4	130.7	54.1	18.8
GPT-4o	1.0	58.7	30.9	311.3	201.0	53.5	9.6
GPT-4	2.9	235.2	87.4	172.3	615.5	243.7	31.5
GPT-4-Turbo	1.8	81.2	54.8	259.9	266.7	76.7	16.6
GPT-4.1-mini	1.0	78.1	29.9	183.6	161.5	69.8	9.7
GPT-4o-mini	1.0	73.4	29.3	370.7	292.1	77.0	10.7
GPT-o4-mini	4.2	164.8	127.4	404.6	503.9	168.2	42.5
GPT-03-mini	4.9	156.2	145.9	177.7	412.3	172.1	44.5
GPT-3.5-Turbo	1.1	51.1	32.8	163.9	199.9	52.7	9.9
Claude-3-7	3.9	231.6	115.5	414.0	457.7	203.0	37.2
Open-source							
Huge Models							
DeepSeek-V3	4.4	169.3	131.2	554.1	461.5	197.2	44.1
DeepSeek-R1	8.1	162.9	242.1	567.1	328.0	134.2	77.4
Qwen3 235B	11.0	180.3	328.8	642.8	440.7	168.9	108.3
Large Models							
LLaMA-4 Maverick	1.2	98.9	37.2	124.2	282.7	118.3	13.6
LLaMA-4 Scout	2.3	105.3	69.1	172.0	300.4	103.6	24.6
DeepSeek-R1-70B	1.5	96.0	44.4	168.0	312.5	99.0	15.2
LLaMA-3-70B	1.5	93.2	45.4	193.7	257.4	76.3	15.0
Medium Models							
Qwen QwQ 32B	0.8	77.2	24.0	186.0	253.1	74.2	8.4
Qwen3-30B	2.3	73.9	68.2	164.4	297.4	83.7	24.6
Gemma3-27B-instruct	2.8	140.9	82.8	297.4	535.0	133.9	26.3
Small Models							
Gemma-2-9B	0.5	116.9	15.4	105.3	283.1	111.2	5.1
LLaMA-3-8B	1.4	155.3	42.6	212.1	573.4	144.9	13.5
Qwen-3-14B	4.1	165.1	123.6	520.0	357.3	176.5	45.0
Gemma3-12B-instruct	3.1	184.7	94.4	400.2	487.9	195.1	33.5

5063
 5064
 5065
 5066
 5067
 5068
 5069
 5070
 5071
 5072
 5073
 5074
 5075

5076 Table 17: Summary Table showing overall performance (%) across six FL environments under
 5077 Fine-grained (FG) and Goal-oriented (GO) guidance.
 5078

5079 Model	5080 Dermatology		5081 Ultrasound		5082 MRI		5083 Fundus		5084 X-ray		5085 Histopathology	
	5086 FG	5087 GO	5088 FG	5089 GO	5090 FG	5091 GO	5092 FG	5093 GO	5094 FG	5095 GO	5096 FG	5097 GO
Proprietary Models												
GPT-4.1	94.29	88.57	94.29	94.29	100.00	100.00	97.14	94.29	100.00	100.00	94.29	94.29
GPT-4o	71.43	65.71	68.57	65.71	71.43	68.57	74.29	74.29	71.43	68.57	65.71	62.86
GPT-4	65.71	51.43	65.71	62.86	71.43	65.71	68.57	65.71	71.43	65.71	54.29	51.43
GPT-4-Turbo	68.57	74.29	65.71	74.29	71.43	77.14	68.57	77.14	71.43	77.14	57.14	65.71
GPT-4.1-mini	88.57	82.86	85.71	77.14	88.57	77.14	88.57	77.14	88.57	77.14	85.71	80.00
GPT-4o-mini	68.57	57.14	71.43	71.43	77.14	74.29	77.14	71.43	77.14	74.29	65.71	60.00
GPT-o4-mini	85.71	74.29	85.71	82.86	91.43	85.71	88.57	85.71	91.43	85.71	77.14	68.57
GPT-03-mini	65.71	60.00	60.00	62.86	71.42	74.29	68.57	74.29	71.42	74.29	71.43	68.57
GPT-3.5-Turbo	25.71	31.43	25.71	34.29	25.71	34.29	25.71	34.29	25.71	34.29	25.71	31.43
Claude-3-7-Sonnet	51.42	57.14	54.29	54.29	57.14	57.14	57.14	57.14	57.14	57.14	51.43	57.14
Open-source Models												
Huge Models												
DeepSeek-V3	85.71	80.00	91.43	85.71	94.29	88.57	94.29	88.57	94.29	88.57	91.43	88.57
DeepSeek-R1	42.86	42.85	42.86	42.86	42.86	42.86	42.86	42.86	42.86	42.86	42.86	42.86
Qwen3-235B	42.86	42.85	42.86	42.86	42.86	42.86	42.86	42.86	42.86	42.86	42.86	42.86
LLaMA-4 Maverick	71.43	74.29	88.57	82.86	94.29	85.71	91.43	85.71	94.29	85.71	77.14	71.43
LLaMA-4 Scout	74.29	77.14	85.71	74.28	85.71	74.29	85.71	74.28	85.71	74.28	80.00	77.14
Large Models												
DeepSeek-R1-70B	45.71	42.86	74.28	68.57	74.29	68.57	71.43	68.57	74.28	68.57	42.86	42.86
LLaMA-3-70B	54.29	57.14	68.57	68.57	71.43	71.43	71.43	71.43	71.43	71.43	54.29	60.00
Medium-sized Models												
Qwen QwQ 32B	91.43	85.71	85.71	80.00	88.57	82.86	88.57	82.86	88.57	82.86	85.71	82.86
Qwen3-30B	45.71	45.71	45.71	45.71	48.57	45.71	48.57	45.71	48.57	45.71	45.71	45.71
Gemma3-27B-instruct	14.29	14.29	14.29	14.29	14.29	14.29	14.29	14.29	14.29	14.29	14.29	14.29
Small Models												
Gemma-2-9B	51.43	51.43	51.43	48.57	51.43	48.57	51.43	48.57	51.43	48.57	57.14	54.29
LLaMA-3-8B	65.71	65.71	60.00	60.00	62.86	60.00	62.86	60.00	62.86	60.00	65.71	65.71
Qwen-3-14B	42.86	40.00	28.57	25.71	28.57	25.71	28.57	25.71	28.57	25.71	42.86	40.00
Gemma3-12B-instruct	14.29	14.29	14.29	14.29	14.29	14.29	14.29	14.29	14.29	14.29	14.29	14.29

5104 reports the average normalized score. GPT-4.1 achieves the highest performance (94.29) under
 5105 both guidance types, demonstrating strong generalization across all subtasks. Smaller models like
 5106 Gemma3-12B-instruct significantly underperform (14.29), especially when tasks require coherent
 5107 execution across multiple stages. Open-source models such as DeepSeek-V3 and LLaMA-4 Maverick
 5108 exhibit competitive performance with proprietary models under fine-grained prompts but show mild
 5109 performance decline in goal-oriented execution.

5110 Table 12 evaluates LLM agents' capability to simultaneously plan and generate independent scripts
 5111 for each subtask in the Ultrasound environment without any explicit or implicit guidance on the
 5112 workflow or availability of tools. It is expected to plan the entire process for completion of each
 5113 subtask as well as write scripts for completing the tasks. This setup is more challenging than the
 5114 previous table. A uniform drop in performance is observed across all models, regardless of type
 5115 or size. Most top proprietary models, such as the GPT-4 series, drop to a common score of 42.86,
 5116 indicating reliance on guided execution for complex task planning. Mid- and small-scale models like
 5117 Claude-3-7, Gemma3-12B, and Qwen-3-14B perform poorly, with scores as low as 14.29 to 31.43,
 5118 demonstrating the importance of tools for domain-specific and robust task understanding.

5119 Table 13 presents evaluation in the **MRI environment**, following the same structure. GPT-4.1
 5120 again leads with a perfect score (100.00) under both guidance types. A general trend of better
 5121 performance under fine-grained guidance than goal-oriented guidance is maintained across most
 5122 models. Open-source large-scale models such as DeepSeek-V3 and LLaMA-4 Maverick narrow the
 5123 performance gap significantly, achieving scores above 85 under fine-grained guidance. Models with
 5124 weaker subtask handling like Gemma3-12B-Instruct remain consistently poor performers, struggling
 5125 to follow multi-step instructions even in highly structured MRI tasks.

5126 Table 14 presents the evaluation of proprietary and open-source LLM agents deployed in the **Fundus**
 5127 **environment**. The table demonstrates that proprietary models such as GPT-4.1 and GPT-4.1-mini
 5128 achieve near-perfect scores across both guidance styles, indicating robust task execution capabilities.
 5129 GPT-4.1 achieves the highest overall fine-grained score (97.14) and maintains a strong goal-oriented
 score (94.29), suggesting high generalization capacity even with minimal instruction. In contrast,

5130 smaller models like GPT-3.5-Turbo and Gemma3-12B-instruct exhibit major limitations, particularly
 5131 under goal-oriented prompting, often failing multiple subtasks and scoring below 35.
 5132

5133 Among open-source models, DeepSeek-V3 and LLaMA-4 Maverick lead performance under both
 5134 guidance types, with fine-grained scores above 90 and goal-oriented scores above 85. These models
 5135 close the gap with top proprietary agents, showcasing the progress of the open-source ecosystem.
 5136 However, performance drops significantly in lightweight models such as Qwen-3-14B and Gemma3-
 5137 12B-instruct, which perform well only on the most basic subtasks and fail to coordinate complex
 5138 operations under goal-driven conditions.

5139 Table 15 presents results for open-source and proprietary LLM agents in the **XRay environment**.
 5140 Proprietary systems remain strongest: GPT-4.1 achieves ceiling performance (5/5 on all sub-tasks). A
 5141 second tier follows with GPT-o4-mini, GPT-4.1-mini, GPT-4o-mini, GPT-4-Turbo, GPT-4, extbfGPT-
 5142 4o, and GPT-o3-mini. Claude-3-7 shows moderate performance, while GPT-3.5-Turbo demonstrates
 5143 substantially weaker performance.

5144 Open-source models narrow the gap in this modality. DeepSeek-V3 and LLaMA-4 Maverick approach
 5145 the top proprietary tier, with LLaMA-4 Scout and Qwen QwQ 32B delivering competitive results.
 5146 Lower-performing models include DeepSeek-R1, Qwen3-235B, and Gemma3-27B-instruct. Among
 5147 smaller models, LLaMA-3-8B exceeds the performance of Gemma-2-9B and Qwen-3-14B.

5148 Stage-wise behaviour matches other environments: strong models consistently solve Client Selection
 5149 and Federated Training (5/5, 5/5), whereas weaker models falter on Data Pre-processing and Label
 5150 Harmonization, yielding frequent 0/5 or 1/5. Fine-grained guidance generally improves Overall
 5151 scores relative to goal-oriented guidance, confirming the benefit of explicit stepwise supervision for
 5152 X-Ray workflows. Overall, all the tables reveal two key insights: (1) proprietary models consistently
 5153 outperform open-source ones across both settings, (2) fine-grained prompting benefits all models but
 5154 especially weaker ones. More insightful discussion on the results can be found in Appendix D.3.

5155 5156 D.2 DISCUSSION ON TIME-EFFICIENCY

5157 Table 16 compares the average time taken (in seconds) by each agent across the seven subtasks (S1–S4,
 5158 C1–C3) in the pipeline. GPT-4.1 is among the fastest overall, particularly in inference-heavy subtasks
 5159 like S1 and S4. Open-source models such as Qwen3-235B and DeepSeek-R1 exhibit significantly
 5160 higher latency, especially in complex subtasks like C2, where times range from approximately 550 to
 5161 640 seconds. Lightweight models such as Qwen QwQ 32B and Gemma-2-9B complete tasks much
 5162 faster but at the cost of performance, as seen in the other tables. This table complements the prior
 5163 performance evaluations by highlighting the efficiency–performance tradeoff, which is critical for
 5164 real-world federated deployments.

5165 We have conducted a comparison of **time–efficiency vs. performance** for each agent role (S1, C1,
 5166 S2, C2, C3, S3, S4) across model families. Overall, we observe the following:

5167 C2 (data prep) and C3 (label harmonization) dominate wall-clock time for almost every model.
 5168 S1/S2/S3/S4 are comparatively light; differences here are smaller and rarely drive total runtime. The
 5169 best choices balance high stage success and short C2/C3 times. Agent-wise takeaways (cross-model):

5170 (i) S1 (server task extraction/broadcast). Times are uniformly small. Fastest include Gemma-2-9B
 5171 (0.5s), QwQ-32B (0.8s), GPT-4o/4o-mini/4.1-mini (1.0s). This stage won’t bottleneck overall runtime,
 5172 so one should prefer models with higher downstream success rather than saving fractions of a second
 5173 here.

5174 (ii) C1 (client selection). A moderate cost stage. GPT-3.5-Turbo (51.1s) and QwQ-32B (77.2s)
 5175 are among the fastest; GPT-4.1 (64.8s) and GPT-4.1-mini (78.1s) are also efficient. Very large
 5176 open-source models (e.g., Qwen3-235B 180s) are slower without clear gains.

5177 (iii) S2 (approval/coordination). Also light in terms of time complexity. Gemma-2-9B (15.4s),
 5178 QwQ-32B (24.0s), GPT-4o/4o-mini/4.1-mini (29–30s) are quickest.

5179 (iv) C2 (data prep / cleaning). One of the two big time sinks. Fastest include Gemma-2-9B (105s) and
 5180 LLaMA-4 Maverick (124s); GPT-3.5 (164s), Qwen3-30B (164s), LLaMA-3-70B (194s), QwQ-32B
 5181 (186s) are solid. GPT-4.1 (302s) and huge open-source (DeepSeek-V3 554s; Qwen3-235B 643s) are

Table 18: User instruction samples mapped to their ground-truth federated learning algorithms. Each instruction encodes a distinct FL requirement such as class-imbalance mitigation, adaptive optimization, heterogeneous-architecture personalization, prototype-based collaboration, or domain generalization and the corresponding correct algorithm is shown in the rightmost column.

Instr. #	User instruction or requirement	Correct Algorithm
1	Train a federated learning model using an algorithm designed to mitigate both inter-client and intra-client class imbalance while still producing a strong global model.	FedLC
2	Train a federated learning model that supports a dynamic gradient adjustment scheme, allowing the learning rate to adapt based on client updates and training dynamics.	FedOpt
3	Train personalized federated learning models where each client maintains a distinct architecture. Use server-side knowledge distillation to enable joint learning while preventing client drift.	FedMD
4	Train personalized federated learning models where raw parameters cannot be exchanged. Instead, allow clients to exchange only class-centroid embeddings for collaboration.	FedProto
5	Train a federated domain-generalization model that learns domain-invariant representations across clients, enabling strong performance on unseen out-of-distribution clients.	FedSR

slower. LLaMA-4 Maverick and QwQ-32B are strong Pareto options (good success, reasonable C2 time).

(v) C3 (label harmonization). The other major time sink and the hardest stage. Standout: GPT-4.1 (131s)—both fast and high success. Next tier includes QwQ-32B (253s) and LLaMA-3-70B (257s), which are respectable; GPT-4o (201s) is faster than many but weaker on Label Harmonization accuracy. GPT-4 (616s) and huge open-source (e.g., DeepSeek-V3 462s) are slow here.

(vi) S3 (algorithm selection). Lightweight. GPT-3.5 (52.7s), GPT-4o (53.5s), GPT-4.1 (54.1s) are quickest; QwQ-32B (74s) is not far behind. This stage rarely determines end-to-end time.

(vii) S4 (training trigger/monitor). Very small across models. Gemma-2-9B (5.1s) is fastest; QwQ-32B (8.4s), GPT-4o/4.1-mini (9–10s) are close. Not a driver of total latency.

We summarize the overall recommendations based on our experiments below:

Best overall (reliability & time): GPT-4.1 with exceptional C3 time (130.7s) and top success. Best

We summarize the overall recommendations based on our experiments below:

Best overall (reliability & time): GPT-4.1 with exceptional C3 time (130.7s) and top success. Best open-source Pareto: Qwen QwQ 32B with 186s for C2 and 253s for C3 with strong success; or LLaMA-4 Maverick if faster C2 is needed (124s). Budget/latency-focused orchestration: GPT-4.1-mini or GPT-4o-mini (But need to keep in mind the success drop on C3). It is advisable to avoid very large open-source for time-critical runs unless one specifically needs open-source + the higher success of DeepSeek-V3 (and can pay the time cost).

D.3 DISCUSSION ON CLIENT SELECTION, REASONING VS NON-REASONING MODELS AND FAILURE MODES:

Qualitative analysis of client selection across modalities. Figures 10-34 present the qualitative agentic performance in the *Client Selection* stage under three clinical modalities, *viz.*, **skin cancer** (dermatology), **histopathology** (breast cancer detection), and **X-Ray** (pneumonia detection) and contrast *non-thinking/reasoning* and *thinking/reasoning* LLM agents. Across all settings, the figures illustrate *when/how* the server approves or declines prospective clients for federated training. For non-thinking agents (e.g., Figs. 10-12; 21-25; 28-30), the selection is typically concise: the model applies eligibility checks and emits a binary decision (approve/decline) with minimal justification. This often highlights crisp gating on dataset relevance to the target task, basic quality constraints, and coarse client readiness.

5238 Table 19: FL algorithm choices per user instruction (see Table 18) for each model. [] denotes no
 5239 valid algorithm returned.

Model	Instr. 1	Instr. 2	Instr. 3	Instr. 4	Instr. 5
Ground Truth	FedLC	FedOpt	FedMD	FedProto	FedSR
GPT-4.1	FedLC	FedOpt	FedMD	FedProto	FedSR
GPT-4o	FedLC	FedDyn	[]	CCVR	FedIIR
GPT-4	FedLC	FedDyn	FedMD	CCVR	FedSR
GPT-4-Turbo	FedLC	FedOpt	[]	FedProto	FedIIR
GPT-4.1-mini	FedLC	FedDyn	[]	FedProto	FedSR
GPT-4o-mini	FedLC	FedDyn	[]	CCVR	FedIIR
GPT-o4-mini	FedLC	FedOpt	[]	FedProto	FedIIR
GPT-03-mini	FedLC	FedOpt	[]	FedProto	FedIIR
GPT-3.5-Turbo	FedProx	FedOpt	[]	CCVR	[]
Claude-3-7-Sonnet	FedLC	FedOpt	[]	[]	[]
DeepSeek-V3	FedLC	FedOpt	[]	FedProto	FedSR
DeepSeek-R1	[]	FedDyn	[]	CCVR	[]
Qwen3 235B	FedProx	FedDyn	[]	[]	FedProx
LLaMA-4 Maverick	FedProx	FedOpt	FedGen	FedProx	FedProx
LLaMA-4 Scout	FedProx	FedOpt	FedGen	FedProx	FedProx
DeepSeek-R1-70B	[]	FedOpt	[]	CCVR	[]
LLaMA-3-70B	FedProx	FedOpt	[]	FedProx	FedProx
Qwen QwQ 32B	FedLC	FedOpt	FedMD	FedProto	FedSR
Qwen3-30B	[]	[]	[]	FedProto	[]
Gemma3-27B-instruct	FedProx	FedDyn	[]	[]	[]
Gemma-2-9B	FedLC	FedDyn	[]	[]	[]
LLaMA-3-8B	FedProx	FedDyn	[]	FedProto	[]
Qwen-3-14B	[]	[]	[]	[]	[]
Gemma3-12B-instruct	FedProx	FedDyn	[]	[]	[]

5266
 5267
 5268 **Impact of using thinking/reasoning agents** For **thinking/reasoning** agents (e.g., Figs.13-17,
 5269 26-27, 31-34), the server-facing rationale becomes more elaborate. These figures show richer criteria
 5270 such as finer judgements about class balance, labeling consistency, or potential contribution to global
 5271 convergence before issuing approve/decline decisions. While this often results in clearer, auditable
 5272 justifications, it can also introduce overhead: Fig. 16 exemplifies *overthinking*, where extended
 5273 deliberation adds verbosity without changing the final decision. Taken together, the sequences suggest
 5274 a trade-off: explicit reasoning improves transparency and sometimes catches subtle issues, but may
 5275 reduce efficiency and occasionally distract from the primary selection objective.

5276
 5277 **Failure modes: hallucination and task drift.** Figures 18-19 document characteristic **hallucinations**
 5278 during client selection with skin cancer datasets. In one case, the model drifts to an *irrelevant*
 5279 *task*, attempting to solve something other than client eligibility; in another, it answers in *Russian*,
 5280 a response channel misaligned with the specified instruction and downstream system expectations.
 5281 Such behaviors indicate vulnerability to prompt misinterpretation and context leakage even at the
 5282 pre-training data curation stage. The remaining thinking-model traces (e.g., Fig 20) demonstrate suc-
 5283 cessful recoveries where the agent returns to the approval/decline protocol after structured reasoning.

5284
 5285 **Consistency across datasets and tasks.** Across **histopathology** (breast cancer) and **X-Ray** (pneu-
 5286 monia) examples, we observe the same qualitative patterns: non-thinking models provide fast,
 5287 rule-like triage; thinking models surface nuanced justifications but are susceptible to verbosity and
 5288 occasional digressions. The figures collectively map the decision boundary between acceptance and
 5289 rejection anchored in dataset/task alignment and basic quality signals while exposing two practical
 5290 risks for agentic selection: (i) *over-elaboration*, which inflates latency without added value, and
 5291 (ii) *hallucination/task drift*, which can misroute the pipeline if not caught by server-side validation.
 5292 These qualitative insights complement the quantitative tables, clarifying *how* different prompting
 5293 regimes lead to the observed approval/decline outcomes in federated client onboarding.

5292 Table 20: Impact of FL algorithm selection and data preprocessing correctness on downstream model
 5293 performance.
 5294

5295 FL algorithm selection	5296 Comments	5297 Data pre	5298 Accuracy	5299 Precision	5300 Recall	5301 F1 Score	5302 Round no.
5296 <input type="checkbox"/>	5297 Defaulting to FedAvg	5298 <input checked="" type="checkbox"/>	5299 57.6488	5300 58.0859	5301 57.8158	5302 57.9505	5303 69
5296 <input checked="" type="checkbox"/>	5297 Chosen algorithm: FedProx	5298 <input checked="" type="checkbox"/>	5299 72.0668	5300 71.9638	5301 72.0871	5302 72.0254	5303 83
5296 <input checked="" type="checkbox"/>	5297 Chosen algorithm: FedLC	5298 <input checked="" type="checkbox"/>	5299 76.7989	5300 76.8456	5301 76.2180	5302 76.5305	5303 98
5300 <input type="checkbox"/>	5301 Defaulting to FedAvg	5302 <input checked="" type="checkbox"/>	5303 63.7697	5304 64.1144	5305 63.4596	5306 63.7853	5307 64
5300 <input checked="" type="checkbox"/>	5301 Chosen algorithm: FedProx	5302 <input checked="" type="checkbox"/>	5303 75.0048	5304 75.5315	5305 73.9155	5306 74.7148	5307 91
5300 <input checked="" type="checkbox"/>	5301 Chosen algorithm: FedLC	5302 <input checked="" type="checkbox"/>	5303 83.4788	5304 83.1265	5305 83.5065	5306 83.3161	5307 91

5300 D.4 FEDERATED TRAINING PERFORMANCE

5303 To assess whether the chosen algorithm actually improves federated learning performance rather
 5304 than merely satisfying the Training-start checklist, we evaluate models far beyond the Training-
 5305 Start Verification metric. To validate this hypothesis, we run full end-to-end federated learning
 5306 experiments, not just the setup phase. We present five different user instructions (covering traditional
 5307 global FL, personalized FL, and Federated Domain Generalization) and their corresponding ground-
 5308 truth algorithms in Table 18, and we report the performance of all LLMs on these five instructions in
 5309 Table 19. These results show that for Instruction 1, some LLMs incorrectly select FedProx instead of
 5310 FedLC, while others return no algorithm at all. The performance on Instruction 2 is also interesting,
 5311 as several models latch onto the word *dynamic* and wrongly select FedDyn instead of FedOpt. We
 5312 next perform a systematic analysis for Instruction 1, i.e., when the user issues the instruction:

5313 *“Train a federated learning model using an algorithm designed to mitigate both
 5314 inter-client and intra-client class imbalance while still producing a strong global
 5315 model.”*

5317 We evaluate all agentic systems for this condition across the entire Federated workflow. Across the
 5318 40-algorithm repository integrated in FEDAGENTBENCH, we observe that some agents correctly
 5319 select FedLC, the only algorithm explicitly designed for class-imbalance mitigation. Some agents
 5320 incorrectly choose FedProx, which regularizes client drift but does not address class imbalance.
 5321 Others return no algorithm, which results in a fallback to FedAvg, the baseline Federated Learning
 5322 algorithm.

5323 Full experimental results (Appendix Tables 19 and 20 as well as Fig. 8) confirm that the algorithm
 5324 choice indeed affects the final FL performance and convergence, not only the Training-start metric.
 5325 To isolate contributing factors, we compare performance trajectories under two conditions: with and
 5326 without a successful data-preprocessing step, and with correct, incorrect, or absent algorithm selection.
 5327 We assume that the client selection and label harmonization step is performed successfully for this,
 5328 else the system will throw intermediate error and the agents would not be able to reach the final step.
 5329 The accuracy curves in Fig. 8 and the ablation in Table 20 show that:

- 5331 Agents that correctly select FedLC (highlighted in red) *i.e.*, GPT-4.1, GPT-4o, GPT-4, GPT-
 5332 4-Turbo, GPT-4.1-mini, GPT-4o-mini, GPT-o4-mini, GPT-O3-mini, Claude-3-7-Sonnet,
 5333 DeepSeek-V3, Qwen QwQ 32B, Gemma-2-9B consistently achieve the highest accuracy,
 5334 precision, recall, and F1
- 5335 Agents that choose FedProx *i.e.* GPT-3.5-Turbo, Qwen3 235B, LLaMA-4 Maverick,
 5336 LLaMA-4 Scout, LLaMA-3-70B, Gemma3-27B-instruct, LLaMA-3-8B, Gemma3-12B-
 5337 instruct perform moderately better than naive FedAvg, but substantially weaker than FedLC
- 5338 Agents that return no algorithm, *i.e.* DeepSeek-R1, DeepSeek-R1-70B, Qwen3-30B, Qwen-
 5339 3-14B defaulting to FedAvg, perform the worst and fail to handle class imbalance.

5341 The three-panel subplot in Fig. 8 further illustrates that overall performance reduces when the
 5342 preprocessing step fails, affecting all agentic systems. The performance improves for the agents
 5343 in red that correctly preprocess; and subplot 8 (c) shows full performance gains when all agents
 5344 successfully complete preprocessing. In all these cases, we find that the correct algorithmic choice
 5345 of FedLC performs better than FedProx which is incorrectly chosen by some LLMs, which is again
 5346 better than defaulting to FedAvg.

5346 Together, these results demonstrate that **FEDAGENTBENCH** does not rely solely on superficial
 5347 “training start” checks. Instead, we validate the **actual downstream effectiveness** of agent decisions
 5348 including algorithm selection via full-pipeline FL training runs, revealing meaningful differences in
 5349 final performance.

5351 E FUTURE WORK

5353 Our failure-mode analysis highlights several limitations of current LLM agents that offer opportunities
 5354 for improving future agent design and prompting strategies in the following ways:

- 5356 **1. Domain-specific reasoning limitations:** Errors arising from insufficient domain-specific
 5357 reasoning, particularly in tasks such as dermatology label harmonization or ultrasound
 5358 dataset selection, suggest the need for domain-aware agents. Future extensions may integrate
 5359 medical ontologies, specific vocabularies, or lightweight domain adapters to ensure that
 5360 LLM agents reason over clinically valid label and task structures.
- 5361 **2. Challenges with multi-step operations:** Many agents struggled with multi-step operations,
 5362 frequently skipping essential preprocessing actions or performing them in the wrong order.
 5363 This motivates the development of structured prompting templates that enforce explicit
 5364 stepwise execution, checklist-style progress tracking, and intermediate self-verification
 5365 before tool invocation (Chen et al., 2025a). Such structure may reduce the tendency of
 5366 agents to shortcut or collapse multi-stage tasks.
- 5367 **3. Overconfidence and shortcutting:** We observed systematic overconfidence and shortcutting
 5368 where models produced plausible but incorrect outputs rather than expressing uncertainty.
 5369 Incorporating uncertainty-aware behaviors such as confidence reporting, contrastive eval-
 5370 uation of alternative outputs, consistency checks, and self-reflection frameworks across
 5371 multiple reasoning paths may mitigate hallucinations in structured FL operations.
- 5372 **4. Lack of workspace grounding:** Hallucinations and task-type mismatches indicate that
 5373 agents often reasoned without grounding their decisions in the actual client workspace.
 5374 Future research could explore: (i) prompting with explicit instructions to avoid relying on
 5375 prior knowledge and instead use only the information provided via prompts, descriptions,
 5376 or task files, and (ii) workspace-grounded decision pipelines that require agents to inspect
 5377 dataset descriptions, directory structures, and tool metadata before committing to actions.
- 5378 **5. Need for adaptive prompting:** Our results show that fine-grained prompting substantially
 5379 improves performance on complex tasks, whereas high-level prompting is sufficient for
 5380 simpler tasks. This points toward adaptive prompting mechanisms, where the system
 5381 dynamically adjusts prompt granularity through prompt optimization strategies, verification
 5382 strictness, and agent role specialization based on the predicted complexity of each FL
 5383 sub-task (Trivedi et al., 2025; Qu et al., 2025; Ramnath et al., 2025).

5383 Beyond prompt- and agent-level improvements, two broader system-level directions emerge from our
 5384 analysis:

5386 **Phase-specific LLM routing:** One promising direction is the development of phase-specific LLM
 5387 routing systems that dynamically select the most suitable agent or model for each FL sub-task.
 5388 Given the heterogeneous performance of LLMs across phases such as label harmonization and
 5389 client selection, an intelligent routing layer could substantially improve reliability and efficiency by
 5390 leveraging the strengths of different agents.

5391 **Reinforcement learning-based reasoning:** Another promising direction is the integration of
 5392 reinforcement learning-based reasoning models Zhang et al. (2025); Singh et al. (2025). RL-guided
 5393 refinement loops could enable agents to learn task-specific decision policies, such as resolving label-
 5394 ing conflicts, planning multi-step preprocessing pipelines, or selecting appropriate FL algorithms
 5395 using verifiable, workspace-grounded signals. Such adaptive, feedback-driven reasoning may miti-
 5396 gate several observed failure patterns, especially those involving multi-step planning and semantic
 5397 grounding.

5398 Together, these directions open pathways for designing more reliable, grounded, and domain-adapted
 5399 LLM agents capable of robustly orchestrating real-world federated learning workflows.

5400 **F DETAILED INSIGHTS FROM THE BENCHMARK**

5401
 5402 We summarize our observations below, providing clear reasoning and interpretation of the agents'
 5403 behaviors:

5404
 5405 **1. Task–Dataset Alignment Requires Abstract Semantic Reasoning**

5406 A consistent source of failure, especially in client selection and label harmonization, is the
 5407 inability of many agents to reliably match task semantics with the correct dataset types.
 5408 Even when tool outputs clearly specify modality or anatomy, weaker agents struggle to infer,
 5409 for example, that brain tumor classification should ignore MRI segmentation datasets.

5410 These mistakes reflect a deeper issue. The reasoning step requires both:

5411 (a) interpreting the task description, and

5412 (b) mapping it to a dataset or label schema with differing granularity. We observe that large
 5413 reasoning chains frequently drift semantically, leading to inclusion of irrelevant datasets or
 5414 omission of required ones.

5415 **For example:** (i) In our benchmark, agents must infer that a task such as “*brain tumor classification*”
 5416 requires **MRI classification** datasets and not similarly named **MRI segmentation**
 5417 datasets even though both correspond to brain tumors.

5418 (ii) They must correctly interpret the semantics of disease labels, e.g., mapping terms like
 5419 “*melanocytic lesion*,” “*malignant melanoma*,” or “*melanoma in situ*” into the appropriate
 5420 canonical classes.

5421 (iii) They must extract task intent from descriptions such as “*multi-class breast lesion*
 5422 *detection from ultrasound images*,” identifying the modality, anatomy, and task type without
 5423 explicit cues.

5424 (iv) They must resolve ambiguous or partially informative metadata, such as recognizing
 5425 that a dataset on breast ultrasound dataset maybe unsuitable for an ultrasound classification
 5426 workflow despite keyword matches as the modality of the datasets is histopathology instead
 5427 of ultrasound.

5428 These abilities require conceptual understanding and multi-hop semantic inference, which
 5429 many current models struggle to perform reliably.

5430 **2. Fine-Grained Prompts Reduce Reasoning Drift**

5431 Across all environments, **structured prompting consistently improves success rates**. Fine-
 5432 grained prompts constrain the reasoning space by enforcing a deterministic step order, *i.e.*,
 5433 identify the task, list candidates, filter, verify, and justify, thereby reducing opportunities
 5434 for hallucination. Goal-oriented prompts, by contrast, allow unconstrained reasoning drift,
 5435 causing: hallucinated directories, incorrect class lists, misinterpreted dataset schemas,
 5436 premature tool invocation. This effect is pronounced in Label Harmonization, where even
 5437 small deviations in reasoning lead to incomplete or inconsistent mappings and so we have to
 5438 provide the LLMs with examples to map fine-grained classes to broader categories in the
 5439 fine-grained prompting.

5440 This challenge also becomes **pronounced in multi-step planning**, where several models
 5441 struggle to follow the required instruction sequence and frequently deviate from the provided
 5442 overall workflow. Instead of using the available tools to retrieve information from dataset
 5443 folders or algorithm description files, weaker agents often rely on prior knowledge, skip
 5444 essential steps, fabricate missing details, or even attempt to recreate tools that have already
 5445 been supplied - behaviours that lead to unstable and incorrect reasoning.

5446 In our work, we observe several concrete cases where **agents ignore the tools explicitly**
 5447 **provided for the task**. **For example**, even though the *selfclean* tool is available to perform
 5448 dataset cleaning, and dedicated file-reading and file-moving tools are provided to inspect
 5449 and reorganize dataset directories, some agents often skip these tools entirely. Instead, they
 5450 attempt to manually script file operations from scratch thereby hallucinating paths, misusing
 5451 Python syntax, or relying on incomplete domain-specific prior knowledge, which leads to
 5452 errors or incomplete outputs.

5453 In multiple instances, the agent fabricates commands such as `mv *.jpg cleanedimages/`
 5454 or invents non-existent directories like `/data/clean/` rather than invoking the correct tool
 5455 designed for this purpose. These behaviours underscore the difficulty models face in multi-
 5456 step planning: even when a reliable tool exists, the agent may fail to recognize its relevance,

5454 misuse it, or attempt to re-create its functionality, resulting in unstable or incorrect pipeline
 5455 execution.

5456 3. Large Models Often Overthink and Are Not Always More Reliable

5457 Interestingly, **reasoning depth does not scale monotonically with model size**. Open-
 5458 weight mid-scale models such as Qwen QwQ-32B and LLaMA-4 Scout often outperform
 5459 models 2–7× larger across multiple environments. A recurring pattern we observe is that
 5460 larger models engage in excessive “over-thinking” and speculative reasoning that ultimately
 5461 breaks the workflow.

5462 **For example**, in the client selection stage (as illustrated in Figs 10–31), some larger models
 5463 repeatedly re-interpret simple rules, spending 20–30 lines debating a binary decision. In
 5464 some other cases, they still fail to follow the required output template, even if they identify the
 5465 correct dataset. In several cases, the agent returns long explanations or nested justifications
 5466 instead of the precise string format expected by the benchmark (e.g., Approved. Prepare for
 5467 training or the exact canonical algorithm name), causing downstream stages to fail due to
 5468 template mismatches.

5469 Similarly, for FL algorithm selection, certain large models correctly infer the intended
 5470 algorithm but embed it inside a paragraph or speculative rationale instead of returning the
 5471 clean pre-specified output, making it unusable in subsequent phases.

5472 This pattern reflects a deeper reliability issue: larger models often generate unnecessarily
 5473 long reasoning chains, hallucinate intermediate interpretations, or override their own
 5474 correct conclusions, whereas mid-sized models tend to follow instructions more faithfully.
 5475 Ultimately, reliability in this benchmark depends less on model size and more on instruction-
 5476 following discipline, consistent template adherence, and robust grounding in tool-based
 5477 workflows.

5478 4. Workspace-specific Grounding Failures Are a Major Source of Error

5479 Many preprocessing steps require precise grounding in file-system realities: verifying folder
 5480 structures, checking formats, validating the existence of files, and generating correct paths.
 5481 Agents often fail because:

- 5482 (a) they hallucinate paths that resemble pretrained-distribution patterns,
- 5483 (b) they ignore tool outputs that contradict their prior reasoning,
- 5484 (c) they overwrite correct tool results with incorrect guesses,
- 5485 (d) they shortcut multi-step verification procedures.

5486 These behaviours illustrate how current LLMs often prioritize their internal generative
 5487 expectations of how datasets should look over the ground-truth symbolic information
 5488 provided by tools.

5489 A related failure pattern appears prominently in the dataset and algorithm selection stages,
 5490 where agents disregard the datasets explicitly provided to them and instead rely on prior
 5491 knowledge from pretraining.

5492 **For instance**, when given a fixed list of client datasets for skin cancer detection, several
 5493 models ignore the actual available options and instead return well-known public datasets
 5494 such as *ISIC 2018*, *ISIC 2019*, or *ISIC 2020*, even if these datasets are not part of that
 5495 particular setting and are never shown to the agent through tools.

5496 A similar issue arises in the MRI environment, where some agents confidently select external
 5497 datasets purely because they recognize these names from pretraining, despite the fact that
 5498 they are not included anywhere in our simulated clients in that particular scenario.

5499 The same pattern appears during Federated Learning algorithm selection: agents occasionally
 5500 propose algorithms such as *FedConsist*, *FedOptimizer*, or other variants that do not exist
 5501 in our provided algorithm list. These behaviors highlight a strong tendency to fall back
 5502 on pretrained “world knowledge” rather than grounding decisions in the actual symbolic
 5503 inputs provided by the environment, thereby leading to systematic errors, hallucinations,
 5504 and mismatches in the selection stages.

5505 5. Label Harmonization Requires Multi-Hop Semantic Reasoning and domain-specific 5506 knowledge

5507 Label harmonization in medical datasets requires multi-hop semantic reasoning and a
 5508 degree of domain-specific clinical knowledge, especially in healthcare contexts where label
 5509 granularity carries diagnostic meaning.

5508 For the binary skin-lesion task in Figure 6, the agent must understand, for example, that
 5509 “Basal Cell Carcinoma,” “Squamous Cell Carcinoma,” and “Melanoma” are all malignant
 5510 entities, while “Nevus,” “Seborrheic Keratosis,” and “Dermatofibroma” are benign. This
 5511 distinction is rarely explicit in raw dataset labels and must be inferred through medical
 5512 knowledge.

5513 To harmonize these correctly, an agent must:

- 5514 (1) infer which fine-grained labels represent malignant cancers,
- 5515 (2) identify which labels represent benign lesions, and
- 5516 (3) consolidate partially overlapping taxonomies across datasets.

5517 This requires multi-hop reasoning steps such as: Nevus → benign lesion → map to Benign,
 5518 or Basal Cell Carcinoma → skin cancer subtype → Malignant, as well as understanding
 5519 that multiple malignant subtypes must collapse into the same canonical class. Current
 5520 LLMs often lack adequate grounding in medical modalities and terminologies (or they rely
 5521 on incomplete or noisy priors), which explains why some models sometimes misclassify
 5522 “Atypical Nevus” as malignant or treat “Seborrheic Keratosis” as a cancer subtype. To
 5523 perform reliable harmonization across institutions, agents must be conditioned with domain-
 5524 specific information either through lightweight medical knowledge retrieval during the
 5525 workflow, integrating structured medical taxonomies, attaching domain-specific adapters, or
 5526 augmenting prompts with concise clinical definitions of relevant disease categories. Without
 5527 such conditioning, the agent’s harmonization decisions rely solely on general-purpose
 5528 pretrained semantic priors, which are insufficient for accurate clinical label alignment and
 5529 multi-hop medical label consolidation, leading to cascading errors in downstream FL stages.
 5530 All these patterns provide the first systematic view of why current LLM agents struggle even
 5531 before facing real-world FL complexity, and offer concrete directions for developing more
 5532 reliable agent reasoning systems.

5534 G PRIVACY ANALYSIS OF HARMONIZED LABELS AND METADATA

5535 Our benchmark’s contribution lies in system-level automation and task performance evaluation, not
 5536 in proving privacy guarantees. However, since FedAgentBench utilizes harmonized labels and some
 5537 form of metadata exchange across clients, below, we rigorously analyze the privacy implications of
 5538 these harmonized labels and transmitted metadata.

5541 G.1 MUTUAL INFORMATION ANALYSIS

5542 Let X be the original dataset at a client, and $M = f(X)$ represent the harmonized labels and metadata
 5543 extracted from the local dataset X , where f includes only non-identifying structural information and
 5544 label taxonomies. In practice, f is a projection or generalization map (e.g., mapping “melanoma” and
 5545 “BCC” both to “malignant”). To quantify potential data tracing risk, we use Mutual Information (MI):
 5546

$$5547 \text{MI}(X; M) = H(X) - H(X|M)$$

5549 where H is the Shannon entropy.

5551 To guarantee minimal traceability:

$$5553 \text{MI}(X; M) \leq \delta, \quad \delta \rightarrow 0$$

5555 Proof:

- 5557 • By designing the function f (harmonization process), we ensure maximal entropy in
 $H(X|M)$.
- 5559 • Assume f maps multiple distinct datasets $X_i \in \mathcal{X}$ to a similar M . Let $|\mathcal{X}| \gg |\mathcal{M}|$. This
 5560 introduces significant ambiguity, thus:

$$5561 H(X|M) \approx H(X)$$

5562 which implies:

5563
$$\text{MI}(X; M) \approx 0$$

5564 Hence, tracing original data through metadata is theoretically negligible.

5566 **G.2 DIFFERENTIAL PRIVACY (DP) PROOF**

5568 We formalize DP guarantees.

5570 Let \mathcal{A} be a randomized mechanism (e.g., gradient updates with Gaussian noise), and D, D' two
5571 neighboring datasets differing by one record. \mathcal{A} satisfies (ϵ, δ) -DP if:

5573
$$\Pr(\mathcal{A}(D) \in S) \leq e^\epsilon \Pr(\mathcal{A}(D') \in S) + \delta, \quad \forall S \subseteq \text{Range}(\mathcal{A})$$

5575 **Proof Outline:**5576 • If Gaussian noise $\mathcal{N}(0, \sigma^2)$ is added to updates during training:

5577
$$\mathcal{A}(D) = \nabla f(D) + \mathcal{N}(0, \sigma^2)$$

5579 • For mechanism sensitivity Δ , noise variance σ^2 satisfies:

5580
$$\sigma \geq \frac{\Delta \sqrt{2 \ln(1.25/\delta)}}{\epsilon}$$

5583 thus rigorously satisfying DP conditions.

5585 **G.3 K-ANONYMITY ANALYSIS**5586 Let \mathcal{C} be the set of clients. Metadata M ensures k-anonymity if each metadata description transmitted
5587 from a client $m \in M$ is generalized such that it matches at least k indistinguishable clients:

5590
$$\forall m \in M, \quad |\{c \in \mathcal{C} : f(X_c) = m\}| \geq k$$

5591 **Proof:**5593 • By metadata generalization, f is designed such that distinct datasets yield identical or highly
5594 similar metadata.
5595 • Given $|\mathcal{C}| \gg k$, the number of clients per metadata class is enforced:

5596
$$|\{c \in \mathcal{C} : f(X_c) = m\}| \geq k$$

5598 thus rigorously satisfying k-anonymity.

5600 **G.4 PRIVACY-UTILITY TRADE-OFF**5602 Define utility U as the expected accuracy of the trained model, and privacy loss ϵ as above. We have:

5604
$$U(\epsilon) = \mathbb{E}[\text{Acc}(M_\epsilon)] \quad \text{with} \quad \frac{dU}{d\epsilon} > 0$$

5606 implying greater privacy (lower ϵ) results in lower accuracy.5608 **Theoretical Bound:**5609 • Utility degradation due to noise addition (DP) or generalization (k-anonymity) is bounded
5610 by:

5612
$$|U(\epsilon) - U(0)| \leq O\left(\frac{1}{\epsilon}\right)$$

5613 This rigorous mathematical analysis demonstrates that harmonized labels and metadata transmission
5614 in **FedAgentBench** can achieve stringent privacy guarantees with negligible traceability risks, aligning
5615 with formal **differential privacy** and **k-anonymity** standards.

```

5616
5617
5618
5619
5620
5621
5622
5623 Running AgentPipelines/run.py...
5624
5625 Human Requirements: I want to train skin cancer classification model. Only use datasets that have skin cancer classes
5626 -----
5627
5628 communication_1_content Task: Skin cancer classification
5629 Modality: Image
5630
5631 Clients, please respond with the name of the selected dataset that contains skin cancer classes suitable for this classification task.
5632 Selector Content: **Dataset Name** : DDI_skin_dataset
5633 //////////////////////////////////////////////////
5634
5635 communication_2_content Approved. Prepare for training
5636 Selector Content: **Dataset Name** : skinL2_dataset
5637 //////////////////////////////////////////////////
5638
5639 communication_2_content Approved. Prepare for training
5640 Selector Content: **Dataset Name** : ISIC_2020
5641 //////////////////////////////////////////////////
5642
5643 communication_2_content Approved. Prepare for training
5644 Selector Content: **Dataset Name** : PH2Dataset
5645 //////////////////////////////////////////////////
5646
5647 communication_2_content Approved. Prepare for training
5648 Selector Content: **Dataset Name** : Dermnet
5649 //////////////////////////////////////////////////
5650
5651 communication_2_content Approved. Prepare for training
5652 Selector Content: **Dataset Name** : ISIC2018_HAM10000
5653 //////////////////////////////////////////////////
5654
5655 communication_2_content Approved. Prepare for training
5656 Selector Content: no dataset: The provided dataset "skin_disease_classification_kaggle"
5657 " contains classes for skin diseases such as acne, eye bags, and redness, but it does
5658 not include any skin cancer classes. Therefore, it is not suitable for the skin cancer
5659 classification task.
5660 //////////////////////////////////////////////////
5661 communication_2_content Client not needed for the task
5662
5663
5664
5665
5666
5667
5668
5669

```

```

5670
5671
5672
5673
5674 Selector Content: no dataset. The provided dataset "Monkeypox_Skin_Image_Dataset" cont
5675 ains skin disease classes such as Chickenpox, Measles, Monkeypox, and Normal, but it d
5676 oes not have skin cancer classes suitable for the skin cancer classification task.
5677 //////////////////////////////////////////////////////////////////
5678 communication_2_content Client not needed for the task
5679 Selector Content: **Dataset Name** : Dermquest
5680 //////////////////////////////////////////////////////////////////
5681 communication_2_content Approved. Prepare for training
5682 Selector Content: **Dataset Name** : augmented_skin_condition_dataset_kaggle
5683 This dataset contains a skin cancer class named "Carcinoma" suitable for the skin can
5684 cer classification task.
5685 //////////////////////////////////////////////////////////////////
5686 communication_2_content Approved. Prepare for training
5687 Selector Content: **Dataset Name** : fitzpatrick17k
5688 //////////////////////////////////////////////////////////////////
5689 communication_2_content Approved. Prepare for training
5690 Selector Content: **Dataset Name** : Dermis
5691 //////////////////////////////////////////////////////////////////
5692 communication_2_content Approved. Prepare for training
5693 Selector Content: **Dataset Name** : ISIC_2017
5694 //////////////////////////////////////////////////////////////////
5695 communication_2_content Approved. Prepare for training
5696 Selector Content: **Dataset Name** : Mednode
5697 //////////////////////////////////////////////////////////////////
5698 communication_2_content Approved. Prepare for training
5699 Selector Content: **Dataset Name** : ISIC_2024
5700 //////////////////////////////////////////////////////////////////
5701 communication_2_content Approved. Prepare for training
5702 Selector Content: **Dataset Name** : Derma7P
5703 //////////////////////////////////////////////////////////////////
5704 communication_2_content Approved. Prepare for training
5705 Selector Content: **Dataset Name** : skin_disease_kaggle_dataset
5706 This dataset contains skin cancer classes such as Basal Cell Carcinoma (BCC) and Melan
5707 oma, suitable for skin cancer classification.
5708
5709 //////////////////////////////////////////////////////////////////
5710 communication_2_content Approved. Prepare for training
5711 Selector Content: **Dataset Name** : skin_disease_kaggle_dataset
5712 This dataset contains skin cancer classes such as Basal Cell Carcinoma (BCC) and Melan
5713 oma, suitable for skin cancer classification.
5714
5715 //////////////////////////////////////////////////////////////////
5716 communication_2_content Approved. Prepare for training
5717

```

Figure 11: Client Selection with **skin cancer datasets** for **non-thinking/reasoning** models. It shows when/how the server approves clients for training whereas when/how it declines.

5720
5721
5722
5723

```

5724
5725
5726
5727
5728
5729
5730 Selector Content: **Dataset Name** : ISIC_2019
5731 //////////////////////////////////////////////////////////////////
5732 communication_2_content Approved. Prepare for training
5733 Selector Content: no dataset
5734 None of the available datasets include skin cancer classes (e.g., melanoma, basal cel
5735 l carcinoma, squamous cell carcinoma). Therefore, there is no suitable dataset for th
5736 e given task.
5737 //////////////////////////////////////////////////////////////////
5738 communication_2_content Client not needed for the task
5739 Selector Content: **Dataset Name** : PAD_UFES_20
5740 //////////////////////////////////////////////////////////////////
5741 communication_2_content Approved. Prepare for training
5742 Selector Content: **Dataset Name** : ISIC_2016
5743 //////////////////////////////////////////////////////////////////
5744 communication_2_content Approved. Prepare for training
5745 Selector Content: **Dataset Name** : skin_disease_3_class
5746 This dataset includes the class "basal cell carcinoma," which is a type of skin cance
5747 r.
5748 //////////////////////////////////////////////////////////////////
5749 communication_2_content Approved. Prepare for training
5750 Selector Content: no dataset
5751 The only available dataset ("Dermatology_tabular dataset") is non-image (tabular) and
5752 does not include dermatology images or specific skin cancer classes, so it is not su
5753 itable for a skin cancer classification model.
5754 //////////////////////////////////////////////////////////////////
5755 communication_2_content Client not needed for the task
5756 Selector Content: no dataset
5757 The provided dataset ("skin-infection-disease-datasaaet") covers only infectious skin
5758 conditions (cellulitis, impetigo, athlete's foot, nail fungus, ringworm, cutaneous la
5759 rva migrans, chickenpox, shingles) and does not include any skin cancer classes.
5760 //////////////////////////////////////////////////////////////////
5761 communication_2_content Client not needed for the task
5762 Selector Content: no dataset
5763 None of the available datasets include skin cancer classes; the only dataset provided
5764 (Monkeypox_Skin_Image_Dataset) contains Chickenpox, Measles, Monkeypox, and Normal c
5765 ategories, which do not cover any form of skin cancer.
5766 //////////////////////////////////////////////////////////////////
5767
5768
5769
5770
5771
5772
5773
5774
5775
5776
5777

```

Figure 12: Client Selection with **skin cancer datasets** for **non-thinking/reasoning** models. It shows when/how the server approves clients for training whereas when/how it declines.

```

5778
5779
5780
5781
5782 Selector Content: **Dataset Name** : skin_disease_3_class
5783 //////////////////////////////////////////////////
5784
5785 communication_2_content Since the client has returned a dataset named "skin_disease_
5786 _3_class", it indicates that they have provided a dataset.
5787
5788 Therefore, the response should be: "Approved. Prepare for training".
5789 Selector Content: The dataset description indicates that the dataset is a tabular d
5790 ataset containing clinical features for diagnosing skin diseases, which doesn't mat
5791 ch the modality of Dermatology Images. Therefore, this dataset is not suitable for
5792 the task of Skin Cancer Detection using Dermatology Images.
5793
5794 Since I have to respond with the name of the selected dataset, I will check if ther
5795 e's any other dataset that matches the task and modality.
5796
5797 //////////////////////////////////////////////////
5798
5799 communication_2_content Since the client has returned a dataset, although it's not
5800 suitable for the task, the client has still responded with a dataset. However, the
5801 response indicates that the dataset is not suitable.
5802
5803 The correct interpretation of the task is to respond based on whether the client ha
5804 s returned any dataset. Since the client has returned a dataset (even though it's n
5805 ot suitable), the response should be based on the presence of a dataset.
5806
5807 Therefore, the response should be: "Approved. Prepare for training".
5808 Selector Content: **Dataset Name** : skin-infection-disease-datasaaet
5809 //////////////////////////////////////////////////
5810
5811 communication_2_content Since the client has returned a dataset named "skin-infecti
5812 on-disease-datasaaet", I will return the approval message.
5813
5814 "Approved. Prepare for training"
5815 Selector Content: **Dataset Name** : Monkeypox_Skin_Image_Dataset
5816 //////////////////////////////////////////////////
5817
5818 communication_2_content Since the client has returned a dataset, **Dataset Name** :
5819 Monkeypox_Skin_Image_Dataset, we can infer that the client has indeed provided a d
5820 ataset for the task.
5821
5822 Therefore, the response should be: "Approved. Prepare for training".
5823 Selector Content: **Dataset Name** : Dermquest
5824 //////////////////////////////////////////////////
5825
5826 communication_2_content Since the client has returned a dataset named "Dermquest",
5827 the response should be: "Approved. Prepare for training".
5828
5829 Therefore, the output is: Approved. Prepare for training
5830
5831

```

Figure 13: Client Selection with **skin cancer datasets** for **thinking/reasoning** models. It shows when/how the server approves clients for training whereas when/how it declines.

5832
 5833
 5834
 5835
 5836
 Selector Content: **Dataset Name** : DermatP
 ///
 5837
 communication_2_content
 <think>
 Okay, the user provided a response with the dataset name "DermatP". Let me check the instructions again. The task is to determine the appropriate reply based on whether the client has returned one or more datasets. The instructions say if there's one or more datasets, I should respond with "Approved. Prepare for training". If none, then "Client not needed for the task".
 So, the user mentioned "DermatP" which is a dataset name, since they specified one dataset here, that meets the condition of having at least one dataset. Therefore, the correct response should be "Approved. Prepare for training". I need to make sure not to add anything else. Just the exact message as per the instructions. Let me double-check that the dataset name is present, which it is. Alright, that's all.
 </think>
 Approved. Prepare for training.
 Selector Content: **Dataset Name** : skin_disease_kaggle_dataset
 ///
 5842
 communication_2_content
 <think>
 Okay, let me see. The user provided a response where the client mentioned a dataset called "skin_disease_kaggle_dataset". The instructions say that if the client returns one or more datasets, I should approve and prepare for training; if not, send the other message.
 So first, check if there's a dataset name listed. Here, the dataset name is clearly given. The format specifies to look for **Dataset Name**. The user input has exactly that. So the condition is met. The correct response is "Approved. Prepare for training".
 I need to make sure there's no trick here. Maybe check if the dataset is valid? But the problem doesn't mention validating the dataset's existence or correctness, just whether it's present. Since the name is present, the answer is straightforward. No need to consider anything else. Just follow the first instruction.
 </think>
 Approved. Prepare for training.
 Selector Content: **Dataset Name** : no dataset
 The available skin dataset includes skin condition categories like "GROWTH OR MOLE" but does not explicitly include **skin cancer subtypes** (e.g., melanoma, basal cell carcinoma) required for direct detection of skin cancer. Its focus on broader skin issues (e.g., acne, pigmentation problems, hair loss) makes it unsuitable for the specific task of cancer classification.
 ///
 5848
 communication_2_content
 <think>
 Okay, let me parse this step by step. The user provided a response from a client agent in a federated learning setup. The client mentioned that their dataset, "skin dataset," includes categories like "GROWTH OR MOLE" but doesn't explicitly cover specific skin cancer subtypes such as melanoma or basal cell carcinoma. They also noted the dataset focuses more on broader skin issues like acne or hair loss, making it unsuitable for cancer classification.
 The user's instructions are clear: if the client returns one or more applicable datasets, the server should respond with "Approved. Prepare for training." If no suitable dataset is provided, the response should be "Client not needed for the task."
 First, I need to determine if the client's dataset meets the requirements. The key point here is that while the dataset includes some relevant categories (like moles), it lacks explicit subtypes of skin cancer needed for direct detection. The client explicitly states that the dataset is unsuitable for the specific task of cancer classification.
 Therefore, since the provided dataset doesn't fulfill the necessary criteria, the correct response per instructions is to inform the client they aren't needed. The user's guidelines specify that if no applicable dataset is returned, the message should be "Client not needed for the task." Even though the client did mention a dataset, it's not suitable, so it counts as not providing a needed dataset.
 No other datasets are mentioned, so there's no approval needed. The answer should be straightforward based on the client's own assessment of their dataset's inadequacy.
 </think>
 Client not needed for the task

Figure 14: Client Selection with **skin cancer datasets** for **thinking/reasoning** models. It shows when/how the server approves clients for training whereas when/how it declines.

5855
 5856
 5857
 5858
 5859
 5860
 5861
 5862
 5863
 5864
 5865
 .Dataset Name** : no dataset. The available dataset ("Dermatology_tabular dataset") contains only tabular clinical features and no image data, which does not match the requirement for "dermatology image's" needed for skin cancer detection.
 The client has a dataset called "Dermatology_tabular dataset" which they think does not match. In their "Dataset Name", they input "no dataset" possibly as a way to indicate they can't contribute.
 Therefore, if "no dataset" is the dataset name they are returning, then they have "no dataset" as their dataset name, which would mean they have no suitable dataset. Thus "no dataset" would be considered as no dataset, so server's response is "Client not needed".
 Alternatively, their having a dataset: "The available dataset (Dermatology...) exists, but it's not suitable. But in their response, they also have "no dataset" as the dataset name, which might be an error.
 Alternatively, the client's dataset name is: "no dataset", which is their way of indicating no contribution.
 Given that the first line says: Dataset Name: no dataset. The rest explains why.
 Thus, in this case, if the client's returned dataset name is "no dataset", then the count is zero, so return "Client not needed".
 Alternatively, the client actually has a dataset (Dermatology...) and the "no dataset" may be an error.
 Since the user's problem says "available dataset... contains only tabular...", which is a dataset. So the client does have a dataset, but it's unsuitable.
 But per the user's instructions given to the assistant to choose between the two options based on if any dataset was returned, regardless of suitability, then since there is one dataset (even if unsuitable), server says "Approved".
 This is conflicting.
 Alternatively, perhaps "returned one or more datasets" refers to datasets that comply, and if they don't comply, they aren't counted.
 The user's own analysis says that the available dataset doesn't match requirement, so perhaps "returned one or more datasets" that matches the task.
 Given that, the client has no dataset that matches, so they have zero datasets that qualify, thus "Client not needed".
 But the user's instruction explicitly states that if the client has "returned one or more datasets", which may not consider suitability, just existence.
 Hmm. I think there's ambiguity, but based on the user's initial statement, if the client provides any dataset, even unsuitable, the server must respond approved, which would be "Approved. Prepare for training".
 Alternatively, the client's message says "***Dataset Name**: no dataset." which is zero datasets.
 Therefore, the server would return "Client not needed".
 The problem lies in parsing what "Dataset Name: no dataset" means. If "no dataset" is their way of indicating that they can't contribute (so zero datasets), then "Client not needed".

Figure 15: Client Selection with **skin cancer datasets** for **thinking/reasoning** models. It shows when/how the server approves clients for training whereas when/how it declines.

```

5886
5887
5888
5889
5890 Selector Content: **Dataset Name** : skin_disease_3.class
5891 ///////////////////////////////////////////////////
5892 communication_2_content
<think>
5893 okay, let me tackle this. The user provided a response where the dataset name is "skin.disease_3.class". The instruction says that if the client has returned one or more datasets, I should reply with "Approved. Prepare for training". if they didn't return any, then "Client not needed for the task".
5894 so first, I need to check if the client's response includes any datasets. The user input here clearly lists a dataset name, which is "skin.disease_3.class". That counts as one dataset. Since there's at least one dataset mentioned, the correct response should be the approved message. I need to make sure I don't misspell anything and follow the exact phrasing required. Alright, looks straightforward here.
</think>
5895 Approved. Prepare for training
5896 Selector Content: **Dataset Name** : no dataset. The available dataset ("Dermatology tabular dataset") contains only tabular clinical features and no image data, which does not match the requirement for "dermatology images" needed for skin cancer detection
5897 ///////////////////////////////////////////////////
5898 communication_2_content
<think>
5899 okay, let's tackle this step by step. The user provided a response from a client agent in a federated learning setup. The client mentioned that their dataset is named "Dermatology tabular dataset" which only has tabular clinical features and no images. The task requires dermatology images for skin cancer detection.
5900 First, I need to check the instructions given. The user said that if the client returns one or more datasets, the server should reply with "Approved. Prepare for training". However, if there's no dataset that meets the requirements, the response should be "Client not needed for the task".
5901 Looking at the client's message, they explicitly stated that their available dataset doesn't include image data. The requirement is for images, so the dataset doesn't fit. Even though the client provided a dataset, the user's instruction says it's not suitable. According to the instruction, if the client has returned one or more datasets, return the "Approved. Prepare for training". Wait, that seems a bit conflicting. Wait, the user's message says: "If the client has returned one or more datasets, return the : 'Approved. Prepare for training'. If the client has returned no dataset, return the message: 'Client not needed for the task'." So perhaps regardless of whether the dataset matches, if they returned a dataset, the server must say approved, even if the data is wrong?
5902 Wait, that doesn't make much sense. Maybe the user's instruction is that the server is supposed to only check if there's any dataset provided, not whether it's suitable. But looking back, the user's initial instruction to me (the assistant) says to follow that rule. Wait, but the user's own problem here is that the client's dataset isn't meeting the requirement. The user's own analysis is that the dataset doesn't match, so the client shouldn't be needed. But according to the user's given rules, if the client returned a dataset (even if not suitable), you have to say "Approved...". Wait, that's conflicting.
5903 Wait, perhaps there's a misinterpretation here. Let me read again the user's exact instruction on how to respond. The user says:
5904 "If the client has returned one or more datasets, return the : 'Approved. Prepare for training'. If the client has returned no dataset, return the message: 'Client not needed for the task'."
5905 So, the user's own instruction is that as long as the client provided any dataset, regardless of its suitability, the server should respond with "Approved...", but the client's dataset in this case does provide a dataset, but it's not suitable because it lacks images. However, according to the given rules, we must return "Approved..." because a dataset was returned, even though it doesn't fit the task requirements. But that contradicts the actual requirement of needing images. Maybe there's a mistake here, but perhaps I'm supposed to follow strictly the user's instruction.
5906 The user's initial instruction says that the client's response indicates that their dataset doesn't meet the requirement. The client says that their available dataset has no image data, which is needed. The task is to decide whether to approve or not based on the presence of datasets, but according to the user's own rules, presence of a dataset (even non-matching) leads to approval. Wait, that's conflicting. The user wants us to process the client's response according to the given rules, which might have a flaw here.
5907 Wait, perhaps the user's instructions have a condition where even if the client provides a dataset, if it doesn't fit, we should treat it as if no dataset was returned. Because the task requires image data, and the client's dataset doesn't have images, so maybe "no dataset" in the sense of not meeting the task's needs. Wait, but according to the message given, the client does have a dataset (the tabular one) but it's not suitable. The user's own instruction says: "If the client has returned one or more datasets..." so the count is about how many datasets they provided, not whether they are suitable. So in this case, the client did provide a dataset (the Dermatology dataset), so according to the user's instruction, the server should reply "Approved. Prepare for training".
5908

```

Figure 16: Client Selection with **skin cancer datasets** for **thinking/reasoning** models. It shows when/how the server approves clients for training whereas when/how it declines. **This is an instance of typical overthinking**

```

5912
5913
5914
5915
5916
5917
5918
5919
5920 Selector Content: **Dataset Name** : no dataset
5921 Reason: The available dataset "skin_disease_classification.kaggle" contains classes like 'acne', 'eye bags', and 'redness', which are skin conditions rather than skin cancer subtypes (e.g., melanoma, basal cell carcinoma). Lack of malignancy-related classes makes it mismatched to skin cancer detection.
5922 ///////////////////////////////////////////////////
5923 communication_2_content
<think>
5924 okay, let's tackle this. The user provided a response where the client mentioned the dataset "skin_disease_classification.kaggle" isn't suitable for skin cancer detection. The reason given is that the dataset's classes like 'acne', 'eye bags', and 'redness' are skin conditions, not cancer subtypes like melanoma or basal cell carcinoma.
5925 So the task here is to determine the correct reply based on the instructions. The instructions say if the client returns one or more datasets, I should say "Approved. Prepare for training". If they don't return any then the reply should be "Client not needed for the task".
5926 Looking at the client's response, they explicitly stated "no dataset" because the available one doesn't fit. Since they didn't provide a suitable dataset, I need to check if they have any other datasets or if they've declined. Since the dataset they have is mismatched, they aren't providing one that meets the requirement.
5927 Therefore, since there's no approved dataset from the client, the correct response is "Client not needed for the task". I must ensure I'm not missing any hidden dataset mentioned, but the client clearly states "no dataset", so the answer is straightforward.
</think>
5928 Client not needed for the task
5929 Selector Content: **Dataset Name** : ISIC 2019
5930 Reason: The ISIC 2019 dataset explicitly includes dermatology images (dermoscopic and clinical images) with labels for skin conditions like Melanoma, Basal Cell Carcinoma, and Squamous Cell Carcinoma, which are critical for skin cancer detection. The dataset aligns with both the **task** (classification of skin cancers) and **modality** (dermatology images).
5931 ///////////////////////////////////////////////////
5932 communication_2_content
<think>
5933 okay, the user provided a response about the ISIC 2019 dataset. I need to check if they mentioned any datasets. The message clearly states "Dataset Name: ISIC 2019" and gives a reasoning related to skin cancer classification, which matches the task and modality. Since they specified a dataset, the next step is to approve and prepare for training. The instructions say to return "Approved. Prepare for training" if datasets are returned. I should make sure there's no ambiguity here. Yep, looks good. The response meets the criteria, so the correct reply is the approved message.
5934 Approved. Prepare for training
5935 Selector Content: **Dataset Name** : ISIC 2019
5936 Reason: "read_file" arguments": [{"file_path": "/home/chris419/Desktop/0Philly work/AI_agents/W3Berger-2023 Dermatology Image Collection focusing on benign lesions in European populations, the dataset might not be optimal for detecting malignant skin cancers. I recommend considering alternative annotated collections with confirmed diagnostic labels for cancer detection. For now, there are no datasets matching your exact criteria. Please review available options.\n\n"
5937
5938
5939

```

Figure 17: Client Selection with **skin cancer datasets** for **thinking/reasoning** models. It shows when/how the server approves clients for training whereas when/how it declines.

```

5940
5941
5942
5943
5944 communication_2_content Here's a breakdown of how to approach the problem of determining if a string is a valid palindrome, along with code examples in Python:
5945 **Understanding Palindromes**
5946 A palindrome is a sequence of characters that reads the same backward as forward. Examples:
5947
5948     • "nada"
5949     • "racecar"
5950     • "A man, a plan, a canal: Panama" (ignoring case and non-alphanumeric characters)
5951
5952 **Steps to Check for a Palindrome**
5953 1. **Preprocessing (optional):** If you need to handle cases with spaces, punctuation, or capitalization, you'll need to preprocess the string. This generally means:
5954     • Converting the string to lowercase
5955     • Removing non-alphanumeric characters (letters and numbers).
5956 2. **Comparison:** Compare the original (or preprocessed) string with its reverse. There are several ways to do this:
5957     • Two Pointers: Start pointers at the beginning and end of the string. Move them towards the middle, comparing characters at each position. If at any point the characters don't match, it's not a palindrome.
5958     • Reverse the string.
5959 Selector Content: "json"
5960 [
5961     {
5962         "function": {
5963             "name": "read_files",
5964             "description": "Read file contents and return as dictionary\n\nArgs:\n    file_paths: List of file paths to read\n\nReturns:\n    dict: Dictionary with {file_path: file_content} format",
5965             "parameters": {
5966                 "properties": {
5967                     "file_paths": {
5968                         "items": [],
5969                         "type": "array"
5970                     }
5971                 },
5972                 "required": [
5973                     "file_paths"
5974                 ],
5975                 "type": "object"
5976             }
5977             "arguments": {
5978                 "file_paths": [
5979                     "/home/chr16419/Desktop/DPhil work/AI_agents/M3Builder/ReferenceFiles/DataCard/Individul_datasets/dataset_7.json"
5980                 ]
5981             }
5982         }
5983     }
5984 ]
5985
5986
5987
5988
5989
5990
5991
5992
5993

```

Figure 18: Typical example of **hallucination** in Client Selection with **skin cancer datasets** for **thinking/reasoning** models. It shows the model is trying to solve irrelevant task.

```

5963
5964
5965
5966
5967
5968
5969
5970
5971 communication_2_content Как приступы паники могут влиять на работу?
5972 Приступы паники могут сильно влиять на работу, проявляясь различными способами. Вот некоторые из них:
5973 **1. Снижение производительности и концентрации:**
5974
5975     • **Трудности с концентрацией:** Во время приступа паники сложно сосредоточиться на задачах, что приводит к снижению производительности. Мысли могут быть хаотичными, а внимание рассеянным.
5976     • **Снижение скорости работы:** Человек может работать медленнее из-за тревоги и страха.
5977     • **Трудности с принятием решений:** Паника может затуманивать ясность мышления, затрудняя принятие даже простых решений.
5978     • **Задачи-инициаторы:** Тревога может негативно влиять на память, приводя к забывчивости и ошибкам.
5979
5980 **2. Проблемы с последовательностью в выполнении обязанностей:**
5981
5982     • **Пропуски работы:** Страх перед приступом паники на работе может привести к тому, что человек будет часто пропускать работу.
5983     • **Ошибки:** Тревога, связанная с работой, может заставить человека
5984 Selector Content: "json"
5985 [
5986     {
5987         "type": "function",
5988         "function": {
5989             "name": "read_files",
5990             "description": "Read file contents and return as dictionary\n\nArgs:\n    file_paths: List of file paths to read\n\nReturns:\n    dict: Dictionary with {file_path: file_content} format",
5991             "parameters": {
5992                 "properties": {
5993                     "file_paths": {
5994                         "items": [],
5995                         "type": "array"
5996                     }
5997                 },
5998                 "required": [
5999                     "file_paths"
6000                 ],
6001                 "type": "object"
6002             }
6003             "arguments": {
6004                 "file_paths": [
6005                     "/home/chr16419/Desktop/DPhil work/AI_agents/M3Builder/ReferenceFiles/DataCard/individual_datasets/dataset_24.json"
6006                 ]
6007             }
6008         }
6009     }
6010 ]
6011
6012
6013
6014
6015
6016
6017
6018
6019
6020
6021
6022
6023
6024
6025
6026
6027
6028
6029
6030
6031
6032
6033
6034
6035
6036
6037
6038
6039
6040
6041
6042
6043
6044
6045
6046
6047
6048
6049
6050
6051
6052
6053
6054
6055
6056
6057
6058
6059
6060
6061
6062
6063
6064
6065
6066
6067
6068
6069
6070
6071
6072
6073
6074
6075
6076
6077
6078
6079
6080
6081
6082
6083
6084
6085
6086
6087
6088
6089
6090
6091
6092
6093
6094
6095
6096
6097
6098
6099
6100
6101
6102
6103
6104
6105
6106
6107
6108
6109
6110
6111
6112
6113
6114
6115
6116
6117
6118
6119
6120
6121
6122
6123
6124
6125
6126
6127
6128
6129
6130
6131
6132
6133
6134
6135
6136
6137
6138
6139
6140
6141
6142
6143
6144
6145
6146
6147
6148
6149
6150
6151
6152
6153
6154
6155
6156
6157
6158
6159
6160
6161
6162
6163
6164
6165
6166
6167
6168
6169
6170
6171
6172
6173
6174
6175
6176
6177
6178
6179
6180
6181
6182
6183
6184
6185
6186
6187
6188
6189
6190
6191
6192
6193
6194
6195
6196
6197
6198
6199
6200
6201
6202
6203
6204
6205
6206
6207
6208
6209
6210
6211
6212
6213
6214
6215
6216
6217
6218
6219
6220
6221
6222
6223
6224
6225
6226
6227
6228
6229
6230
6231
6232
6233
6234
6235
6236
6237
6238
6239
6240
6241
6242
6243
6244
6245
6246
6247
6248
6249
6250
6251
6252
6253
6254
6255
6256
6257
6258
6259
6260
6261
6262
6263
6264
6265
6266
6267
6268
6269
6270
6271
6272
6273
6274
6275
6276
6277
6278
6279
6280
6281
6282
6283
6284
6285
6286
6287
6288
6289
6290
6291
6292
6293
6294
6295
6296
6297
6298
6299
6300
6301
6302
6303
6304
6305
6306
6307
6308
6309
6310
6311
6312
6313
6314
6315
6316
6317
6318
6319
6320
6321
6322
6323
6324
6325
6326
6327
6328
6329
6330
6331
6332
6333
6334
6335
6336
6337
6338
6339
6340
6341
6342
6343
6344
6345
6346
6347
6348
6349
6350
6351
6352
6353
6354
6355
6356
6357
6358
6359
6360
6361
6362
6363
6364
6365
6366
6367
6368
6369
6370
6371
6372
6373
6374
6375
6376
6377
6378
6379
6380
6381
6382
6383
6384
6385
6386
6387
6388
6389
6390
6391
6392
6393
6394
6395
6396
6397
6398
6399
6400
6401
6402
6403
6404
6405
6406
6407
6408
6409
6410
6411
6412
6413
6414
6415
6416
6417
6418
6419
6420
6421
6422
6423
6424
6425
6426
6427
6428
6429
6430
6431
6432
6433
6434
6435
6436
6437
6438
6439
6440
6441
6442
6443
6444
6445
6446
6447
6448
6449
6450
6451
6452
6453
6454
6455
6456
6457
6458
6459
6460
6461
6462
6463
6464
6465
6466
6467
6468
6469
6470
6471
6472
6473
6474
6475
6476
6477
6478
6479
6480
6481
6482
6483
6484
6485
6486
6487
6488
6489
6490
6491
6492
6493
6494
6495
6496
6497
6498
6499
6500
6501
6502
6503
6504
6505
6506
6507
6508
6509
6510
6511
6512
6513
6514
6515
6516
6517
6518
6519
6520
6521
6522
6523
6524
6525
6526
6527
6528
6529
6530
6531
6532
6533
6534
6535
6536
6537
6538
6539
6540
6541
6542
6543
6544
6545
6546
6547
6548
6549
6550
6551
6552
6553
6554
6555
6556
6557
6558
6559
6560
6561
6562
6563
6564
6565
6566
6567
6568
6569
6570
6571
6572
6573
6574
6575
6576
6577
6578
6579
6580
6581
6582
6583
6584
6585
6586
6587
6588
6589
6590
6591
6592
6593
6594
6595
6596
6597
6598
6599
6600
6601
6602
6603
6604
6605
6606
6607
6608
6609
6610
6611
6612
6613
6614
6615
6616
6617
6618
6619
6620
6621
6622
6623
6624
6625
6626
6627
6628
6629
6630
6631
6632
6633
6634
6635
6636
6637
6638
6639
6640
6641
6642
6643
6644
6645
6646
6647
6648
6649
6650
6651
6652
6653
6654
6655
6656
6657
6658
6659
6660
6661
6662
6663
6664
6665
6666
6667
6668
6669
6670
6671
6672
6673
6674
6675
6676
6677
6678
6679
6680
6681
6682
6683
6684
6685
6686
6687
6688
6689
6690
6691
6692
6693
6694
6695
6696
6697
6698
6699
66100
66101
66102
66103
66104
66105
66106
66107
66108
66109
66110
66111
66112
66113
66114
66115
66116
66117
66118
66119
66120
66121
66122
66123
66124
66125
66126
66127
66128
66129
66130
66131
66132
66133
66134
66135
66136
66137
66138
66139
66140
66141
66142
66143
66144
66145
66146
66147
66148
66149
66150
66151
66152
66153
66154
66155
66156
66157
66158
66159
66160
66161
66162
66163
66164
66165
66166
66167
66168
66169
66170
66171
66172
66173
66174
66175
66176
66177
66178
66179
66180
66181
66182
66183
66184
66185
66186
66187
66188
66189
66190
66191
66192
66193
66194
66195
66196
66197
66198
66199
66200
66201
66202
66203
66204
66205
66206
66207
66208
66209
66210
66211
66212
66213
66214
66215
66216
66217
66218
66219
66220
66221
66222
66223
66224
66225
66226
66227
66228
66229
66230
66231
66232
66233
66234
66235
66236
66237
66238
66239
66240
66241
66242
66243
66244
66245
66246
66247
66248
66249
66250
66251
66252
66253
66254
66255
66256
66257
66258
66259
66260
66261
66262
66263
66264
66265
66266
66267
66268
66269
66270
66271
66272
66273
66274
66275
66276
66277
66278
66279
66280
66281
66282
66283
66284
66285
66286
66287
66288
66289
66290
66291
66292
66293
66294
66295
66296
66297
66298
66299
66300
66301
66302
66303
66304
66305
66306
66307
66308
66309
66310
66311
66312
66313
66314
66315
66316
66317
66318
66319
66320
66321
66322
66323
66324
66325
66326
66327
66328
66329
66330
66331
66332
66333
66334
66335
66336
66337
66338
66339
66340
66341
66342
66343
66344
66345
66346
66347
66348
66349
66350
66351
66352
66353
66354
66355
66356
66357
66358
66359
66360
66361
66362
66363
66364
66365
66366
66367
66368
66369
66370
66371
66372
66373
66374
66375
66376
66377
66378
66379
66380
66381
66382
66383
66384
66385
66386
66387
66388
66389
66390
66391
66392
66393
66394
66395
66396
66397
66398
66399
66400
66401
66402
66403
66404
66405
66406
66407
66408
66409
66410
66411
66412
66413
66414
66415
66416
66417
66418
66419
66420
66421
66422
66423
66424
66425
66426
66427
66428
66429
66430
66431
66432
66433
66434
66435
66436
66437
66438
66439
66440
66441
66442
66443
66444
66445
66446
66447
66448
66449
66450
66451
66452
66453
66454
66455
66456
66457
66458
66459
66460
66461
66462
66463
66464
66465
66466
66467
66468
66469
66470
66471
66472
66473
66474
66475
66476
66477
66478
66479
66480
66481
66482
66483
66484
66485
66486
66487
66488
66489
66490
66491
66492
66493
66494
66495
66496
66497
66498
66499
66500
66501
66502
66503
66504
66505
66506
66507
66508
66509
66510
66511
66512
66513
66514
66515
66516
66517
66518
66519
66520
66521
66522
66523
66524
66525
66526
66527
66528
66529
66530
66531
66532
66533
66534
66535
66536
66537
66538
66539
66540
66541
66542
66543
66544
66545
66546
66547
66548
66549
66550
66551
66552
66553
66554
66555
66556
66557
66558
66559
66560
66561
66562
66563
66564
66565
66566
66567
66568
66569
66570
66571
66572
66573
66574
66575
66576
66577
66578
66579
66580
66581
66582
66583
66584
66585
66586
66587
66588
66589
66590
66591
66592
66593
66594
66595
66596
66597
66598
66599
66600
66601
66602
66603
66604
66605
66606
66607
66608
66609
66610
66611
66612
66613
66614
66615
66616
66617
66618
66619
66620
66621
66622
66623
66624
66625
66626
66627
66628
66629
66630
66631
66632
66633
66634
66635
66636
66637
66638
66639
66640
66641
66642
66643
66644
66645
66646
66647
66648
66649
66650
66651
66652
66653
66654
66655
66656
66657
66658
66659
66660
66661
66662
66663
66664
66665
66666
66667
66668
66669
66670
66671
66672
66673
66674
66675
66676
66677
66678
66679
66680
66681
66682
66683
66684
66685
66686
66687
66688
66689
66690
66691
66692
66693
66694
66695
66696
66697
66698
66699
66700
66701
66702
66703
66704
66705
66706
66707
66708
66709
66710
66711
66712
66713
66714
66715
66716
66717
66718
66719
66720
66721
66722
66723
66724
66725
66726
66727
66728
66729
66730
66731
66732
66733
66734
66735
66736
66737
66738
66739
66740
66741
66742
66743
66744
66745
66746
66747
66748
66749
66750
66751
66752
66753
66754
66755
66756
66757
66758
66759
66760
66761
66762
66763
66764
66765
66766
66767
66768
66769
66770
66771
66772
66773
66774
66775
66776
66777
66778
66779
66780
66781
66782
66783
66784
66785
66786
66787
66788
66789
66790
66791
66792
66793
66794
66795
66796
66797
66798
66799
66800
66801
66802
66803
66804
66805
66806
66807
66808
66809
66810
66811
66812
66813
66814
66815
66816
66817
66818
66819
66820
66821
66822
66823
66824
66825
66826
66827
66828
66829
66830
66831
66832
66833
66834
66835
66836
66837
66838
66839
66840
66841
66842
66843
66844
66845
66846
66847
66848
66849
66850
66851
66852
66853
66854
66855
66856
66857
66858
66859
66860
66861
66862
66863
66864
66865
66866
66867
66868
66869
66870
66871
66872
66873
66874
66875
66876
66877
66878
66879
66880
66881
66882
66883
66884
66885
66886
66887
66888
66889
66890
66891
66892
66893
66894
66895
66896
66897
66898
66899
66900
66901
66902
66903
66904
66905
66906
66907
66908
66909
66910
66911
66912
66913
66914
66915
66916
66917
66918
66919
66920
66921
66922
66923
66924
66925
66926
66927
66928
66929
66930
66931
66932
66933
66934
66935
66936
66937
66938
66939
66940
66941
66942
66943
66944
66945
66946
66947
66948
66949
66950
66951
66952
66953
66954
66955
66956
66957
66958
66959
66960
66961
66962
66963
66964
66965
66966
66967
66968
66969
66970
66971
66972
66973
66974
66975
66976
66977
66978
66979
66980
66981
66982
66983
66984
66985
66986
66987
66988
66989
66990
66991
66992
66993
66994
66995
66996
66997
66998
66999
669999
670000
670001
670002
670003
670004
670005
670006
670007
670008
670009
670010
670011
670012
670013
670014
670015
670016
670017
670018
670019
670020
670021
670022
670023
670024
670025
670026
670027
670028
670029
670030
670031
670032
670033
670034
670035
670036
670037
670038
670039
670040
670041
670042
670043
670044
670045
670046
670047
670048
670049
670050
670051
670052
670053
670054
670055
670056
670057
670058
670059
670060
670061
670062
670063
670064
670065
670066
670067
670068
670069
670070
670071
670072
670073
670074
670075
670076
670077
670078
670079
670080
670081
670082
670083
670084
670085
670086
670087
670088
670089
670090
670091
670092
670093
670094
670095
670096
670097
670098
670099
6700999
6701000
6701001
6701002
6701003
6701004
6701005
6701006
6701007
6701008
6701009
67010010
67010011
67010012
67010013
67010014
67010015
67010016
67010017
67010018
67010019
67010020
67010021
67010022

```

```

5994
5995
5996 communication_2_content <think>
5997     Okay, let me try to figure out how to approach this. The user wants to train a skin cancer detection model using dermatology images. So first, I need to check the available datasets to see which ones are
5998     relevant.
5999     The user mentioned a specific file path: /home/chr16419/Desktop/DPHIL work/AI_agents/M3Builder/ReferenceFiles/DataCard/individual_datasets/dataset_23.json. I should start by reading that file to get the
6000     list of datasets. The function read_files is available for this purpose.
6001     Once I have the dataset descriptions, I'll look for any that mention skin cancer detection or dermatology images. The key is to match both the task (detection) and the modality (dermatology images). If there
6002     are multiple datasets, I'll list them. If none fit, I'll inform the user.
6003     I need to make sure I use the correct syntax for the tool call. The file path is provided, so I'll pass that to the read_files function. Then, based on the response, I can proceed to select the appropriate
6004     datasets.
6005     </think>
6006     {"name": "read_files", "arguments": {"file_paths": ["file:///home/chr16419/Desktop/DPHIL work/AI_agents/M3Builder/ReferenceFiles/DataCard/individual_datasets/dataset_17.json"]}}
6007     Selector Content: <think>
6008     Okay, the user wants to train a skin cancer detection model using dermatology images. Let me start by understanding the requirements. The task is skin cancer detection, and the modality is dermatology images, which likely refers to dermoscopic images or similar medical images of the skin.
6009     First, I need to check the available datasets. The user mentioned a specific file path: /home/chr16419/Desktop/DPHIL work/AI_agents/M3Builder/ReferenceFiles/DataCard/individual_datasets/dataset_17.json. I should use the read_files function to read this file and examine its contents. The dataset descriptions will help determine if any of them are suitable for the task.
6010     I'll call the read_files function with the provided file path. Once I have the dataset information, I'll look for entries where the description mentions skin cancer detection or related terms, and ensure the modality is dermatology images. If there are matching datasets, I'll list their names. If not, I'll inform the user that no suitable dataset was found.
6011     </think>
6012     <tool_call>
6013     {"name": "read_files", "arguments": {"file_paths": ["file:///home/chr16419/Desktop/DPHIL work/AI_agents/M3Builder/ReferenceFiles/DataCard/individual_datasets/dataset_17.json"]}}
6014     communication_2_content <think>
6015     Okay, let me proceed this step by step. The user is working on a skin cancer detection project using dermatology images. They provided a file path to a dataset description file. My job is to check if the dataset is suitable for their task.
6016     First, I need to read the contents of the specified JSON file. The file path is /home/chr16419/Desktop/DPHIL work/AI_agents/M3Builder/ReferenceFiles/DataCard/individual_datasets/dataset_17.json. Using the read_files function, I'll access the dataset descriptions stored there.
6017     Once I have the data, I'll look for any entries that mention "skin cancer detection" in their description. The modality should be dermatology images, which might include terms like "dermoscopic images," "skin lesions," or similar. If there's a match, I'll confirm the dataset's suitability. If not, I'll inform the user that no relevant dataset was found.
6018     I should also check if the dataset's structure and content align with typical requirements for training a detection model, such as having labeled images of skin lesions. If the dataset meets these criteria, the next step is to proceed with training. Otherwise, the user might need to look for alternative datasets or
6019     </think>
6020     okay, the user wants to train a skin cancer detection model using dermatology images. Let me start by understanding the requirements. The task is skin cancer detection, and the modality is dermatology images, which likely refers to dermoscopic images or similar.
6021     First, I need to check the available datasets. The user mentioned a specific file path: /home/chr16419/Desktop/DPHIL work/AI_agents/M3Builder/ReferenceFiles/DataCard/individual_datasets/dataset_10.json. I should use the read_files function to read this file and examine its contents.
6022     By reading the JSON file, I'll look at each dataset's description to see if it matches the task and modality. The goal is to find datasets that contain dermatology images used for skin cancer detection. If the file isn't found or doesn't contain relevant data, I'll need to inform the user.
6023     Once I have the dataset information, I'll filter out those that fit the criteria. If there are no suitable datasets, I'll respond with "no dataset" and explain why. Otherwise, I'll list the matching data set names as instructed.
6024     </think>
6025     <tool_call>

```

Figure 20: Client Selection with **skin cancer datasets** for **thinking/reasoning** models. It shows when/how the server approves clients for training whereas when/how it declines.

```

6026 Human Requirements: I want to train binary breast cancer detection model (i.e., benign and malignant) from histopathology images.
6027 .....  

6028 communication_1_content Task: Binary breast cancer detection (benign vs. malignant)
6029 Modality: Histopathology Images
6030 Clients, please respond with the name of the selected dataset that matches this requirement.
6031 Selector Content: **Dataset Name** : BreakHis_400X
6032 .....  

6033 communication_2_content Approved. Prepare for training.
6034 Selector Content: **Dataset Name** : HER2E
6035 .....  

6036 communication_2_content Approved. Prepare for training.
6037 Selector Content: **Dataset Name** : breast_cancer_segmentation
6038 .....  

6039 communication_2_content Approved. Prepare for training.
6040 Selector Content: no dataset
6041 Explanation: The "BreakHis" dataset is a breast cancer histopathology dataset, but it is focused on multi-class segmentation of histological structures (mitosis, apoptosis, tumor nuclei, etc.), not on binary classification of breast cancer as benign vs. malignant. Therefore, it does not match the requirement for binary breast cancer detection.
6042 .....  

6043 communication_2_content Client not needed for the task.
6044 Selector Content: no dataset
6045 Explanation: The available dataset, "HER2E," contains histopathology images of gastric (stomach) cancer, not breast cancer. Therefore, it does not match the requirement for binary breast cancer detection (benign vs. malignant) using histopathology images.
6046 .....  

6047 communication_2_content Client not needed for the task.
6048 Selector Content: no dataset
6049 Explanation: The available dataset, "breast_cancer_segmentation," contains histopathology images for breast cancer, not breast cancer. Therefore, it does not match the requirement for binary breast cancer detection (benign vs. malignant).

```

Figure 21: Client Selection with **histopathology datasets** for **non-thinking/reasoning** models on **breast cancer detection task**. It shows when/how the server approves clients for training whereas when/how it declines.

```

6048
6049
6050
6051
6052
6053
6054
6055
6056
6057
6058
6059
6060
6061
6062
6063
6064
6065
6066
6067
6068
6069
6070
6071
6072
6073
6074
6075
6076
6077
6078
6079
6080
6081
6082
6083
6084
6085
6086
6087
6088
6089
6090
6091
6092
6093
6094
6095
6096
6097
6098
6099
6100
6101

```

```

"Dataset Name": "No suitable dataset for binary breast cancer detection from histopathology images in the provided file.
////////////////////////////////////////////////////////////////
communication_2_content Client not needed for the task
Selector Content: Based on the file content, the dataset suitable for your task is:
"Dataset Name": "breast_histos
This dataset is a breast cancer histopathology dataset for binary classification. It contains two folders: 'IDC' and 'non-IDC', representing the presence or absence of invasive ductal carcinoma (IDC), wh
ich is a type of breast cancer. This makes it suitable for your task of binary breast cancer detection from histopathology images.
////////////////////////////////////////////////////////////////
communication_2_content Approved. Prepare for training
Selector Content: Based on the content of the json file, we have one dataset that fits the task:
"Dataset Name": "breast_cancer_histos
This dataset contains histopathological images of breast cancer divided into benign and malignant categories, which is suitable for binary breast cancer detection.
////////////////////////////////////////////////////////////////
communication_2_content Approved. Prepare for training
Selector Content: Based on the content of the json file, the dataset available is "gastro_cancer_msi_vs_mss". This dataset is a gastrointestinal histopathology dataset with subfolders representing MSI and MSS classes for classification.
However, this dataset does not suit your requirements as it is not a binary breast cancer detection dataset. It is a gastrointestinal cancer dataset and the classes are MSI and MSS, not benign and malignant which are required for binary breast cancer detection.
Therefore, there is no suitable dataset available for the given task of binary breast cancer detection from histopathology images.
"Dataset Name": "No suitable dataset
////////////////////////////////////////////////////////////////
communication_2_content Client not needed for the task
Selector Content: The dataset available is "ACROBAT_Breast". It contains 4,212 whole slide images (WSIs) from 1,153 patients for registration across H&E and IHC-stained breast cancer tissue sections. However, it is important to note that this dataset includes landmark annotations for registration evaluation and it's not explicitly mentioned whether it has labels for binary breast cancer detection (benign and malignant). Therefore, further investigation might be needed to determine if this dataset is suitable for your task.
////////////////////////////////////////////////////////////////
communication_2_content Approved. Prepare for training
Selector Content: The dataset in the provided JSON file is named "lung_and_colon". However, this dataset is not suitable for the task of binary breast cancer detection as it consists of lung histopathology images, not breast. Therefore, there is no suitable dataset for the given task in the provided JSON file.
"Dataset Name": "No suitable dataset
The reason is that the available dataset "lung_and_colon" is related to lung histopathology, not breast cancer detection.
////////////////////////////////////////////////////////////////
communication_2_content Client not needed for the task.

```

Figure 22: Client Selection with **histopathology datasets for non-thinking/reasoning** models on **breast cancer detection task**. It shows when/how the server approves clients for training whereas when/how it declines.

```

6077
6078
6079
6080
6081
6082
6083
6084
6085
6086
6087
6088
6089
6090
6091
6092
6093
6094
6095
6096
6097
6098
6099
6100
6101

```

```

Human Requirements: I want to train binary breast cancer detection model (i.e., benign and malignant) from histopathology images.
-----
communication_2_content Task: Binary Breast Cancer Detection
Modality: Histopathology Images
Dear clients, please respond with the name of the selected dataset that contains histopathology images for binary breast cancer detection.
Selector Content: The dataset that matches the task of binary breast cancer detection using histopathology images is:
"Dataset Name": "BreastKHS_400X
This dataset consists of histopathological images at 400X magnification. The data is arranged into two folders: 'benign' and 'malignant', which makes it suitable for binary breast cancer detection.
////////////////////////////////////////////////////////////////
communication_2_content Approved. Prepare for training
Selector Content: Based on the content of the dataset file, the available dataset is "HEROHE". This dataset is a breast cancer histopathology dataset used in the Grand Challenge. It contains 300 cases (144 positive, 216 negative) and includes a metadata file providing HER2 expression status. This dataset can be used for binary breast cancer detection, as it contains both positive and negative cases.
////////////////////////////////////////////////////////////////
communication_2_content Approved. Prepare for training
Selector Content: Based on the content of the dataset description file, the suitable dataset for your task is:
"Dataset Name": "breast_cancer_segmentation
This dataset contains 58 H&E stained histopathology images of breast cancer along with corresponding ground truth segmentation masks. It can be used for binary breast cancer detection (i.e., benign and malignant) from histopathology images.
////////////////////////////////////////////////////////////////
communication_2_content Approved. Prepare for training
Selector Content: Based on the information in the dataset description, the dataset "recalining" contains breast cancer histopathology images. However, it supports multi-class segmentation into six categories: mitotic, necrosis, normal, non-tumor, nuclei, tubule, and non-tubule. It does not explicitly mention whether it can be used for binary breast cancer detection (i.e., benign and malignant). Therefore, we may need further investigation or preprocessing to use this dataset for the specific task of binary breast cancer detection.
////////////////////////////////////////////////////////////////
communication_2_content Approved. Prepare for training
Selector Content: Based on the information from the json file, I found a suitable dataset for your task.
"Dataset Name": "SLN_Breast
This dataset consists of H&E-stained whole slide images (WSIs) labeled as positive or negative for metastatic breast carcinoma, which supports binary classification of lymph node metastasis.
////////////////////////////////////////////////////////////////
communication_2_content Approved. Prepare for training
Selector Content: Based on the dataset description, the dataset named "melanoma" contains histopathology images of melanoma, not breast cancer. Therefore, this dataset is not suitable for binary breast cancer detection.
"Dataset Name": "No suitable dataset
The provided dataset is not suitable for the task of binary breast cancer detection because it contains histopathology images of melanoma, not breast cancer.
////////////////////////////////////////////////////////////////
communication_2_content Client not needed for the task.
Selector Content: From the dataset description, the dataset "gastric_cancer" contains H&E-stained histopathological images from human gastric cancer. However, this dataset is not suitable for your task since it does not contain histopathology images for binary breast cancer detection.
Therefore, there is no suitable dataset available for the given task in the provided dataset description.
////////////////////////////////////////////////////////////////
communication_2_content Client not needed for the task.

```

Figure 23: Client Selection with **histopathology datasets for non-thinking/reasoning** models on **breast cancer detection task**. It shows when/how the server approves clients for training whereas when/how it declines.

```

6102 Selector Content: no dataset: The provided dataset "breast_cancer_segmentation" is designed for segmentation tasks, as it contains segmentation masks for breast cancer regions rather than labels for benign vs malignant classification.
6103 //////////////////////////////////////////////////
6104 communication_2_content Client not needed for the task
6105 Selector Content: no dataset: The provided dataset "BreCaHAD" is intended for multi-class segmentation of histological structures rather than binary classification into benign versus malignant.
6106 //////////////////////////////////////////////////
6107 communication_2_content Client not needed for the task
6108 Selector Content: **Dataset Name** : SLN_Breast
6109 //////////////////////////////////////////////////
6110 communication_2_content Approved, Prepare for training.
6111 Selector Content: no dataset: The available dataset "melanoma" is for segmentation of melanoma histopathology images and does not align with the binary classification task (benign vs malignant) for breast cancer detection.
6112 //////////////////////////////////////////////////
6113 communication_2_content Client not needed for the task.
6114 Selector Content: no dataset - The dataset provided ("gastric_cancer") is focused on gastric cancer histopathological images and does not match the task of binary classification for breast cancer detection (benign vs malignant).
6115 //////////////////////////////////////////////////
6116 communication_2_content Client not needed for the task
6117 Selector Content: no dataset: The provided dataset "ovarian_cancer" is not suitable for binary breast cancer detection (benign vs malignant) as it only contains data related to ovarian cancer.
6118 //////////////////////////////////////////////////
6119 communication_2_content Client not needed for the task
6120 Selector Content: no dataset: The provided dataset "breast_histo" is not suitable for binary breast cancer detection (benign vs malignant).
6121 //////////////////////////////////////////////////
6122 communication_2_content Approved, Prepare for training
6123 Selector Content: **Dataset Name** : breast_cancer_histo
6124 //////////////////////////////////////////////////
6125 communication_2_content Approved, Prepare for training
6126 Selector Content: no dataset: The provided dataset "gastro_cancer_ms1_vs_ms2", is a gastrointestinal histopathology dataset classified into MSI and MSS classes, which does not meet the requirement for a breast cancer detection task (benign vs malignant).
6127 //////////////////////////////////////////////////
6128 communication_2_content Client not needed for the task
6129 Selector Content: no dataset: The ACROBAT_Breast dataset is designed for registration across H&E and IHC-stained breast cancer tissue sections and includes landmark annotations for registration evaluation. It does not provide the benign versus malignant labels required for binary breast cancer detection.
6130 //////////////////////////////////////////////////
6131 communication_2_content Client not needed for the task
6132 Selector Content: no dataset: The provided dataset "lung_and_colon" is designed for lung and colon tissue classification (including lung benign, adenocarcinoma, and squamous cell carcinoma) and does not match the requirement for binary breast cancer detection using histopathology images.
6133 //////////////////////////////////////////////////
6134 communication_2_content Client not needed for the task

```

Figure 24: Client Selection with **histopathology datasets for non-thinking/reasoning models on breast cancer detection task**. It shows when/how the server approves clients for training whereas when/how it declines.

H BROADER SOCIAL IMPACT

Positive Societal Impacts: FedAgentBench offers a significant advancement toward democratizing access to federated learning (FL) in healthcare by enabling agentic automation of otherwise labor-intensive tasks such as client selection, data cleaning, label harmonization, and FL algorithm configuration. This shift can particularly benefit healthcare institutions in low- and middle-income countries (LMICs) and rural areas, where hiring skilled data engineers is not feasible. By reducing the human expertise required to deploy and manage FL workflows, our benchmark promotes more equitable participation in collaborative medical AI initiatives. Furthermore, the open-source nature of our framework ensures broad accessibility, fostering reproducibility, community contributions, and educational use.

Negative Societal Impacts: The automation of critical decision-making steps in FL workflows, especially in high-stakes healthcare applications, raises concerns about over-reliance on large language models (LLMs) that may hallucinate or misinterpret complex clinical instructions. Incorrect data preprocessing or label harmonization could propagate errors into downstream training, leading to biased or unreliable models. Additionally, while the framework supports privacy-preserving learning, if misused without proper oversight, it may inadvertently reinforce data quality issues or amplify existing healthcare disparities. Careful human oversight and robust safety mechanisms must accompany such autonomous systems to ensure responsible deployment in real-world medical settings.

I LLM USAGE:

LLMs (GPT-4o and GPT-5) were sometimes used for improving grammar and wording.

```

6156
6157
6158
6159
6160
6161
6162
6163
6164
6165 **Task:** Train a binary classification model for breast cancer detection.
6166 **Modality:** Histopathology images
6167 **Client Agents:** Please respond with the name of the dataset you have that matches this requirement.
6168
6169 Selector Content: **Dataset Name** : BreakHis_400X
6170 //////////////////////////////////////////////////
6171 communication_2_content Approved. Prepare for training.
6172
6173 Selector Content: **Dataset Name:** HEROHE
6174 //////////////////////////////////////////////////
6175 communication_2_content Approved. Prepare for training.
6176
6177 Selector Content: **Dataset Name** : breast_cancer_segmentation
6178 //////////////////////////////////////////////////
6179 communication_2_content This response indicates the client has returned a dataset (breast_cancer_segmentation).
6180
6181 Therefore, the appropriate server response is:
6182 **Approved. Prepare for training**
6183
6184 Selector Content: **Dataset Name**: BreCaHAD
6185 //////////////////////////////////////////////////
6186 communication_2_content Approved. Prepare for training.
6187
6188 Selector Content: **Dataset Name**: SLN_Breast
6189 //////////////////////////////////////////////////
6190 communication_2_content Approved. Prepare for training.
6191
6192 Selector Content: no dataset : The provided dataset is for melanoma segmentation, not breast cancer detection.
6193
6194 //////////////////////////////////////////////////
6195 communication_2_content Client not needed for the task.
6196
6197 Selector Content: no dataset
6198 This dataset contains gastric cancer images and not breast cancer

```

6199 Figure 25: Client Selection with **histopathology datasets** for **non-thinking/reasoning** models on
6200 **breast cancer detection task**. It shows when/how the server approves clients for training whereas
6201 when/how it declines.

6202
6203
6204
6205
6206
6207
6208
6209

```

6210
6211
6212
6213
6214
6215
6216
6217
6218
6219
6220
6221
6222
6223
6224
6225
6226
6227
6228
6229
6230
6231
6232
6233
6234
6235
6236
6237
6238
6239
6240
6241
6242
6243
6244
6245
6246
6247
6248
6249
6250
6251
6252
6253
6254
6255
6256
6257
6258
6259
6260
6261
6262
6263

```

```

Selector Content: **Dataset Name** : ACROBAT_Breast
/////////////////////////////////////////////////////////////////
communication_2_content <think>
Okay, so I need to figure out how to respond to the client's message. The client sent a dataset named ACROBAT_Breast. My task is to determine the appropriate response based on whether they provided a dataset or not.
First, I should check if the client returned one or more datasets. In this case, they did provide a dataset, specifically ACROBAT_Breast. So according to the instructions, if they returned one or more datasets, I should respond with "Approved. Prepare for training."
If they hadn't provided any dataset, I would respond with "Client not needed for the task." But since they did provide one, the first response is the correct one.
I should make sure I'm following the instructions accurately. The client's message clearly states the dataset name, so there's no ambiguity here. Therefore, the response should be "Approved. Prepare for training."
</think>
The client provided the dataset ACROBAT_Breast, so the appropriate response is:
**Approved. Prepare for training.**
Selector Content: The task is binary classification, and the modality is medical imaging. The dataset "lung_and_colon" does not match the requirements because it contains lung tissue images, not breast histopathology images. Therefore, no suitable dataset is available for the specified task and modality.
**Dataset Name**: no dataset
/////////////////////////////////////////////////////////////////
communication_2_content <think>
Okay, so I'm trying to figure out how to respond to the client's message. Let me break it down step by step. The client is involved in a Federated Learning setup, and I'm acting as the server agent. My job is to communicate with the client agents, so I need to understand their response and decide the appropriate reply.
First, the client mentioned that the task is binary classification, and the modality is medical imaging. That gives me some context about what they're working on. They're probably looking to classify medical images into two categories, maybe something like tumor vs. no tumor or benign vs. malignant.
Next, the client talked about a dataset called "lung_and_colon" but pointed out that it doesn't fit their needs. They were expecting breast histopathology images, but this dataset has lung tissue images instead. So, the client is saying that this dataset isn't suitable for their specific task.
The client also mentioned that there's no suitable dataset available for their task and modality. That means they can't provide any dataset that meets the server's requirements. The dataset name they provided is "no dataset," which clearly indicates they don't have an appropriate one to offer.
Now, according to the instructions, if the client returns one or more datasets, I should respond with "Approved. Prepare for training." But if they return no dataset, I need to send "Client not needed for the task." In this case, since the client explicitly stated there's no suitable dataset, the correct response is the latter.
I should make sure my response is clear and follows the guidelines without adding any extra information. It's important to stick to the specified messages to avoid confusion. So, I'll prepare the response accordingly.
</think>
Client not needed for the task.

```

Figure 26: Client Selection with **histopathology datasets** for **thinking/reasoning models** on **breast cancer detection task**. It shows when/how the server approves clients for training whereas when/how it declines.

```

6233
6234
6235
6236
6237
6238
6239
6240
6241
6242
6243
6244
6245
6246
6247
6248
6249
6250
6251
6252
6253
6254
6255
6256
6257
6258
6259
6260
6261
6262
6263

```

```

Human Requirements: I want to train binary breast cancer detection model (i.e., benign and malignant) from histopathology images.
communication_1_content <think>
Okay, so the user wants to train a binary breast cancer detection model using histopathology images. First, I need to figure out what exactly they're asking for. They mentioned binary classification, so it's about distinguishing between benign and malignant tumors. The data modality is histopathology images, which are microscope slides of tissue samples.
Now, I should extract the task and modality from their requirement. The task is binary classification, specifically for breast cancer detection. The modality is histopathology images.
Next, I need to instruct the clients to respond with the dataset name that matches these requirements. I should make sure to clearly state the task and modality so the clients can identify the correct dataset.
I wonder if there are standard datasets for this. Maybe something like the Breast Cancer Histopathology dataset or similar. But the exact name depends on what the clients have available.
I should phrase the instruction clearly, asking them to provide the dataset name that fits the binary classification task on histopathology images. That should help them respond accurately.
</think>
The task is **binary classification** for breast cancer detection, and the modality is **histopathology images**.
please respond with the name of the selected dataset that matches this requirement.
Selector Content: **Dataset Name**: Breast Cancer Histopathology Dataset
/////////////////////////////////////////////////////////////////
communication_2_content <think>
Okay, so I need to figure out how to respond to the client's message in this Federated Learning scenario. The client provided a dataset named "Breast Cancer Histopathology Dataset." My task is to determine whether to approve the dataset for training or to send a message indicating the client isn't needed.
First, I should understand the instructions clearly. If the client returns one or more datasets, the server should respond with "Approved. Prepare for training." If no datasets are returned, the response should be "Client not needed for the task."
Looking at the client's response, they provided the dataset name. That means they have returned at least one dataset. So, according to the instructions, the appropriate response is to approve and prepare for training.
I don't think I need to validate the dataset name itself, just whether it exists. Since it does, the response is straightforward. No need for additional checks unless specified, which it isn't here.
So, I'll go with "Approved. Prepare for training."
</think>
Approved. Prepare for training.
Selector Content: The selected dataset for your task is:
**Dataset Name**: HEROHE
/////////////////////////////////////////////////////////////////
communication_2_content <think>
Okay, so I just received a message from the client about a dataset they've selected. Let me try to understand what they're saying. The dataset name is HEROHE, and they mentioned it's suitable because it contains histopathology images specific to breast cancer. These images are labeled as either positive (malignant) or negative (benign), which is exactly what we need for a binary classification task.
Hm, the client has clearly provided a dataset, so according to the instructions, I should respond with "Approved. Prepare for training." That makes sense because they've met the criteria by selecting a relevant dataset. I don't think I need to do anything else here except acknowledge their selection and give the go-ahead for the next step.
Hm, let me double-check the instructions. If they provided one or more datasets, I return "Approved. Prepare for training." If none, then "Client not needed for the task." Since they did provide HEROHE, the first response is appropriate. I don't see any issues here. It seems straightforward.
I should make sure my response is clear and concise. No need for extra information since they've already given the necessary details. Just the approval and the instruction to prepare for training. Alright, I think I've got it.
</think>

```

Figure 27: Client Selection with **histopathology datasets** for **thinking/reasoning models** on **breast cancer detection task**. It shows when/how the server approves clients for training whereas when/how it declines.

```

6264
6265
6266 Selector Content: **Dataset Name** : no dataset
6267 The dataset available, named "bone-fracture-dataset," contains X-ray images of fractured and non-fractured bones, which does not match the requirement for pneumonia detection in chest X-ray images. Therefore, it is not suitable for the specified task.
6268
6269 communication_2_content Client not needed for the task.
6270 Selector Content: **Dataset Name** : anatomy-in-chest-x-rays-pax-ray
6271
6272 communication_2_content Approved. Prepare for training.
6273 Selector Content: **Dataset Name** : no dataset
6274 The dataset "unifesp-xray-bodypart-classification" is designed for identifying 20 body parts using whole body X-ray images and does not specifically target pneumonia detection in chest X-rays. Therefore, it is not suitable for the task of pneumonia detection from chest X-ray images.
6275
6276 communication_2_content Client not needed for the task.
6277 Selector Content: **Dataset Name** : nth-chest-x-rays-bbox-version
6278
6279 communication_2_content Approved. Prepare for training.
6280 Selector Content: **Dataset Name** : covid19-xray-dataset-train-test-sets
6281
6282 communication_2_content Approved. Prepare for training.
6283 Selector Content: **Dataset Name** : no dataset
6284 The provided dataset "dental-radiography" contains dental X-ray images, which are not suitable for training a pneumonia detection model that requires chest X-ray images. Therefore, there is no appropriate dataset available for this specific task.
6285
6286 communication_2_content Client not needed for the task.
6287 Selector Content: **Dataset Name** : chest-x-ray-pneumonia
6288
6289 communication_2_content Approved. Prepare for training.
6290 Selector Content: **Dataset Name** : no dataset
6291 The dataset provided, "bone-break-classification-image-dataset," focuses on different types of bone fractures in X-ray images, such as elbow, wrist, and shoulder fractures. It does not contain images or data specifically related to pneumonia detection in chest X-rays, which is necessary for the task at hand. Therefore, it is not suitable for training a pneumonia detection model.
6292
6293 communication_2_content Client not needed for the task.
6294 Selector Content: **Dataset Name** : No suitable dataset for the given task exists. The available dataset is for bone fracture classification and not for pneumonia detection in chest XRay images.
6295
6296 communication_2_content Client not needed for the task.
6297 Selector Content: The dataset "spr-x-ray-age-and-gender-dataset" contains chest Xray images to detect age and gender of people. It does not seem to be suitable for pneumonia detection as it does not contain labels related to pneumonia.
6298
6299 communication_2_content Client not needed for the task.
6300 Selector Content: The dataset available to use is "covid19-and-normal-x-ray-dataset-balanced". It contains chest X-ray images balanced across COVID-19 positive and normal cases. However, it does not specifically contain data for pneumonia detection. Therefore, it may not be suitable for your task of pneumonia detection from chest XRay images.
6301
6302 communication_2_content Approved. Prepare for training.
6303 Selector Content: Based on the dataset description from the json file, the dataset "human-bone-fractures-image-dataset-h5mfld" is organized into folders based on fracture type and anatomical location. This dataset does not seem to be suitable for pneumonia detection as it is focused on bone fractures. Therefore, there is no suitable dataset for the given task.
6304
6305 communication_2_content Client not needed for the task.
6306 Selector Content: The dataset available is "osteoporosis-database" which includes knee X-ray images labeled for osteoporosis stages. Unfortunately, this dataset is not suitable for the task of pneumonia detection from chest XRay images.
6307
6308 communication_2_content Client not needed for the task.
6309 Selector Content: The suitable dataset for your requirement is:
6310
6311 communication_2_content Approved. Prepare for training.
6312 Selector Content: The dataset contains chest X-ray images with labels 'Normal' and 'Pneumonia'. It is located at the path: /home/chri6419/Desktop/DPHil work/AI_agents/M3Builder/Dataset/xray_dataset/coronahack-chest-xray-dataset.
6313
6314 You can use this dataset to train your pneumonia detection model.
6315
6316 communication_2_content Approved. Prepare for training.
6317 Selector Content: The dataset available is "datasetname", lung-segment. However, this dataset contains lung segmentation masks and not specifically labeled for pneumonia detection. Therefore, it may not be suitable for your task of pneumonia detection from chest X-ray images.

```

Figure 28: Client Selection with **X-Ray datasets for non-thinking/reasoning models on pneumonia detection task**. It shows when/how the server approves clients for training whereas when/how it declines.

```

6294
6295
6296
6297
6298
6299
6300
6301
6302
6303
6304
6305
6306
6307
6308
6309
6310
6311
6312
6313
6314
6315
6316
6317

```

Figure 29: Client Selection with **X-Ray datasets for non-thinking/reasoning models on pneumonia detection task**. It shows when/how the server approves clients for training whereas when/how it declines.

```

6318
6319
6320
6321
6322
6323
6324
6325
6326
6327
6328
6329
6330
6331
6332
6333
6334
6335
6336
6337
6338
6339

**Dataset Name** : covid19-xray-dataset-train-test-sets
This dataset includes chest X-ray images labeled as normal and pneumonia, which matches your requirement for pneumonia detection from chest XRay Images.
=====
communication_2_content Approved. Prepare for training
Selector Content: Based on the dataset description I found, the dataset available is **Dataset Name**: dental-radiography. However, this dataset includes dental X-ray images, which are not suitable for pneumonia detection from chest XRay Images. Therefore, there is no suitable dataset available for the given task in the provided dataset descriptions.
=====
communication_2_content Client not needed for the task
Selector Content: Based on the dataset descriptions, the dataset that matches the user requirement for pneumonia detection from chest XRay Images is:
**Dataset Name** : chest-x-ray-pneumonia
This dataset contains chest X-ray images labeled as PNEUMONIA or NORMAL, split into respective folders for binary classification tasks.
=====
communication_2_content Approved. Prepare for training
Selector Content: Based on the dataset description, the dataset "bone-break-classification-image-dataset" is not suitable for pneumonia detection from chest XRay images as it includes X-ray images categorized into fracture types such as elbow, wrist, and shoulder fractures. Therefore, there is no suitable dataset for the given task in the provided json file.
=====
communication_2_content Client not needed for the task
Selector Content: Based on the file content, the dataset that matches your requirement for pneumonia detection from chest XRay Images is:
**Dataset Name** : chest-x-ray-pneumonia-covid19-tuberculosis
This dataset includes 7,135 chest X-ray images categorized into Normal, Pneumonia, COVID-19, and Tuberculosis in separate subfolders named after respective classes.
=====
communication_2_content Approved. Prepare for training
Selector Content: Based on the dataset descriptions, I found a suitable dataset for your task:
**Dataset Name** : chest-x-ray-17-diseases
This dataset includes chest X-Ray images for lung disease detection, including a subfolder for Pneumonia, which matches your requirement for pneumonia detection from chest XRay Images.
=====
communication_2_content Approved. Prepare for training
Selector Content: Based on the information from the dataset description, the dataset named "fractatas" is an annotated fracture image dataset. It does not seem to be suitable for pneumonia detection from chest XRay Images since it is focused on fractures, not pneumonia.
Therefore, I would say there is no suitable dataset for the given task in the provided dataset descriptions.
**Dataset Name** : No suitable dataset for pneumonia detection from chest XRay images in the provided dataset descriptions.
=====
communication_2_content Client not needed for the task
Selector Content: Based on the dataset description, the available dataset is "digital-knee-x-ray-images" which is a digital knee osteoarthritis grading dataset. This dataset is not suitable for the task of pneumonia detection from chest XRay Images as it contains knee XRay images, not chest XRay images.
**Dataset Name** : No suitable dataset
The reason is that the available dataset is for knee osteoarthritis grading and not for pneumonia detection from chest XRay images.
=====

6340 Client Selection with X-Ray datasets for non-thinking/reasoning models on pneumonia detection task. It shows when/how the server approves clients for training whereas when/how it declines.
6341
6342
6343
6344
6345
6346
6347
6348
6349
6350
6351
6352
6353
6354
6355
6356
6357
6358
6359
6360
6361
6362
6363
6364
6365
6366
6367
6368
6369
6370
6371

```

Figure 30: Client Selection with **X-Ray datasets for non-thinking/reasoning models on pneumonia detection task**. It shows when/how the server approves clients for training whereas when/how it declines.

```

Human Requirements: I want to train pneumonia detection model from chest XRay images.
=====
communication_1_content
<think>
Okay, the user wants to train a pneumonia detection model using chest X-ray images. Let me break this down. The task here is clearly pneumonia detection, which is a classification task. The modality is medical imaging, specifically X-ray images. So I need to extract the task and modality from their request. The user mentioned "chest XRay Images," so modality is X-ray. The dataset they need should be something like the NIH Chest X-Ray dataset or the pneumonia-specific datasets like the one from Vinbigraph or maybe the dataset from the study by Wang et al. But I should just state the task and modality first as instructed, then prompt the clients to provide the dataset name that fits. Wait, the instructions say to only extract task and modality from the user's message. The user's requirement is to train a model for pneumonia detection using chest X-rays, so the task is pneumonia detection, modality is X-ray images. I need to make sure not to add extra info beyond that. Then, I have to structure the response by stating those two extracted pieces and then instruct the clients to respond with the dataset name. Got it.
</think>

**Task** Pneumonia detection
**Modality** Chest X-ray Images

Please respond with the name of the selected dataset that matches this task and modality.
Selector Content: **Dataset Name** : no dataset
The required dataset for pneumonia detection must contain chest X-ray images labeled for pneumonia vs. normal or other lung conditions. The available "fracture-multi-region-x-ray-data" focuses on bone fractures in regions like limbs and knees, which is unrelated to pneumonia detection in chest X-rays. No suitable dataset exists in the provided options for this task.
=====
communication_2_content
<think>
Okay, let's see. The user is acting as a server agent in a federated learning setup. The client responded that they don't have the right dataset.
The instructions say: if the client returns datasets, say "Approved. Prepare for training". If they don't have any, respond "Client not needed for the task".
The client's response clearly states that the available dataset is about fractures, which isn't suitable for pneumonia detection. So they have no appropriate dataset.
Therefore, the correct response should be "Client not needed for the task".
</think>

Client not needed for the task
Selector Content: **Dataset Name** : no dataset
The provided dataset (bone-fracture-dataset) is focused on bone fracture classification, which does not align with the task of pneumonia detection in chest X-rays. No relevant dataset matching "pneumonia detection" and "chest X-ray" was found in the provided dataset entries.
=====
communication_2_content
<think>
Okay, let me process this. The user provided a response indicating that the dataset named "bone-fracture-dataset" isn't suitable for pneumonia detection in chest X-rays. They mentioned no relevant dataset was found.
The instructions say if the client returns no dataset, I should respond with "Client not needed for the task". Since the user states there's no relevant dataset here, that means the client doesn't have the necessary data for the task. Therefore, the correct response is to inform them they're not needed.
I need to make sure I don't approve training since the dataset isn't appropriate. The key points are the mismatch between the dataset and the task, and the absence of a suitable dataset. So the right message here is "Client not needed for the task".
</think>

Client not needed for the task
=====

6340 Client Selection with X-Ray datasets for thinking/reasoning models on pneumonia detection task. It shows when/how the server approves clients for training whereas when/how it declines.
6341
6342
6343
6344
6345
6346
6347
6348
6349
6350
6351
6352
6353
6354
6355
6356
6357
6358
6359
6360
6361
6362
6363
6364
6365
6366
6367
6368
6369
6370
6371

```

Figure 31: Client Selection with **X-Ray datasets for thinking/reasoning models on pneumonia detection task**. It shows when/how the server approves clients for training whereas when/how it declines.

```

6372
6373
6374
6375
6376
6377
6378
6379
6380
6381
6382
6383
6384
6385
6386
6387
6388
6389
6390
6391
6392
6393
6394
6395
6396
6397
6398
6399
6400
6401
6402
6403
6404
6405
6406
6407
6408
6409
6410
6411
6412
6413
6414
6415
6416
6417
6418
6419
6420
6421
6422
6423
6424
6425

**Dataset Name**: chest-x-ray-lungs-segmentation
/////////////////////////////////////////////////////////////////
communication_2_content Client not needed for the task
Selector Content: **Dataset Name** : coronahack-chest-x-ray-dataset
/////////////////////////////////////////////////////////////////
communication_2_content Approved. Prepare for training
Selector Content: [
  tool_call: [
    {
      "id": "pending",
      "type": "function",
      "function": [
        "name": "read_files"
      ],
      "parameters": [
        "file_paths": [
          "/home/chr16419/Desktop/OPhil work/AI_agents/M3Builder/ReferenceFiles/DataCard/Indvldual_datasets_xray/dataset_19.json"
        ]
      ]
    }
  ]
]

/////////////////////////////////////////////////////////////////
communication_2_content The provided client response only indicates an intent to execute a function ('read_files') and doesn't actually provide any information about returned datasets. Therefore, I can't determine if the client has returned datasets or not.
**Here's what I would need to know:**

* **The output of the 'read_files' function:** This would tell us if any datasets were actually read and handled by the client.
* **Once I have that information, I can give you the appropriate response:**

* **If datasets are returned:** "Approved. Prepare for training"
* **If no datasets are returned:** "Client not needed for the task"

Let me know if you can provide the output of 'read_files'.

```

Figure 32: Client Selection with **X-Ray datasets for thinking/reasoning models on pneumonia detection task**. It shows when/how the server approves clients for training whereas when/how it declines.

```

6401
6402
6403
6404
6405
6406
6407
6408
6409
6410
6411
6412
6413
6414
6415
6416
6417
6418
6419
6420
6421
6422
6423
6424
6425

Selector Content: Based on the output of tool call id "call_mbpr", I found a dataset that matches the task: **human-bone-fractures-image-dataset-hbfmid**: human-bone-fractures-image-dataset-hbfmid includes X-Ray Images organized into folders based on fracture type and anatomical location.
/////////////////////////////////////////////////////////////////
communication_2_content Approved. Prepare for training.
Selector Content: **Chest-x-ray-pneumonia** : This dataset matches the task: Pneumonia Detection, and modality: XRay Images.
/////////////////////////////////////////////////////////////////
communication_2_content Approved. Prepare for training
Selector Content: Based on the provided dataset 3 description, **Dataset Name** : chest-x-ray-lungs-segmentation.
/////////////////////////////////////////////////////////////////
communication_2_content I have received the client response. Based on the message, I see that the client has returned the following information: **Dataset Name**: chest-x-ray-lungs-segmentation. Since the client has returned a dataset, I will respond with the message: "Approved. Prepare for training". Selector Content: My dataset selection strategy suggests that the dataset "coronahack-chest-x-ray-dataset" matches the task: Pneumonia Detection Modality: XRay Images. Therefore, I select this dataset for the client.

**Dataset Name** : coronahack-chest-x-ray-dataset
/////////////////////////////////////////////////////////////////
communication_2_content Approved. Prepare for training.
Selector Content: </tool-use>
{
  "tool_calls": [
    {
      "id": "pending",
      "type": "function",
      "function": [
        "name": "read_files"
      ],
      "parameters": [
        "file_paths": [
          "/home/chr16419/Desktop/OPhil work/AI_agents/M3Builder/ReferenceFiles/DataCard/individual_datasets_xray/dataset_19.json"
        ]
      ]
    }
  ]
}</tool-use>

Please note: No tool call actually made here. Please wait for tool call results.
/////////////////////////////////////////////////////////////////
communication_2_content The client response indicates that the client agent is requesting to make a "read_files" tool call with a single file path: "/home/chr16419/Desktop/OPhil work/AI_agents/M3Builder/ReferenceFiles/DataCard/individual_datasets_xray/dataset_19.json".
Since no actual tool call was made, we should wait for the tool call results. Since this is a "function" type tool call and not an actual dataset submission, the response should not be considered as approved or rejected.
We should respond with: "Please wait for tool call results."

```

Figure 33: Client Selection with **X-Ray datasets for thinking/reasoning models on pneumonia detection task**. It shows when/how the server approves clients for training whereas when/how it declines.

6426
6427
6428
6429 **Dataset Name**: NIH Chest X-ray Dataset
6430 Approved. Prepare for training.
6431 Selector Content: **Dataset Name** : covid19-radiography-database
6432 This dataset is suitable for training a pneumonia detection model as it contains images specifically labeled for pneumonia.
6433
6434 /////////////////////////////
6435 communication_2_content </think>
6436 Okay, so I'm trying to figure out how to respond to the client's message in this Federated Learning setup. Let me break it down step by step.
6437 First, the client provided a dataset named "covid19-radiography-database." They mentioned that this dataset is suitable for training a pneumonia detection model because it contains images specifically labeled for pneumonia.
6438 Now, the instructions say that if the client returns one or more datasets, I should respond with "Approved. Prepare for training." If they return no dataset, I should say "Client not needed for the task."
6439 In this case, the client did provide a dataset, so according to the rules, the appropriate response is "Approved. Prepare for training." I don't need to consider any other factors because the instruction is clear about the response based on the presence of a dataset.
6440 I should make sure my response is exactly as specified without adding any extra information. That way, the client knows to proceed with the training phase.
6441 /////////////////////////////
6442 Approved. Prepare for training.
6443 Selector Content: The task is **pneumonia detection** and the modality is **chest X-ray images**.
6444 The provided dataset is **simple-vs-comminuted-fractures-x-ray-data**, which is focused on bone fracture classification (simple vs. comminuted fractures). This dataset does not contain chest X-ray images relevant for pneumonia detection.
6445 **No suitable dataset** for the given task exists in the provided data.
6446 /////////////////////////////
6447 communication_2_content </think>
6448 Okay, let's try to figure out what the user is asking for here. They're acting as a server agent in a Federated Learning setup, communicating with client agents. The client responded with some specific information.
6449 So, the task is pneumonia detection, which means the goal is to build a model that can look at images and determine if someone has pneumonia. The modality specified is chest X-ray images, so the model will be dealing with X-ray pictures of chests.
6450 Now, the client provided a dataset called "simple-vs-comminuted-fractures-x-ray-data". I need to understand what this dataset contains. From the name, it sounds like it's focused on fractures, specifically distinguishing between simple and comminuted fractures. These are types of bone fractures, so the images are likely X-rays of bones, maybe arms or legs, showing different fracture types.
6451 The user mentioned that this dataset doesn't have chest X-rays, which makes sense because it's about fractures, not lung issues. Since the task is pneumonia detection, which requires images of lungs, this dataset isn't suitable.
6452 The instructions say that if the client provided one or more datasets, the server should return "Approved. Prepare for training." But if there are no suitable datasets, the response should be "Client not needed for the task."
6453 In this case, the client provided a dataset, but it's not relevant to the task. So, the server needs to determine if any of the datasets are suitable. Since none are, the correct response is "Client not needed for the task."
6454 I should make sure I'm following the instructions correctly. The key is whether the dataset is suitable. Even though a dataset was provided, it's not the right type, so the server doesn't need the client for this task.
6455 /////////////////////////////
6456 client not needed for the task.

Figure 34: Client Selection with **X-Ray datasets** for **thinking/reasoning** models on **pneumonia detection task**. It shows when/how the server approves clients for training whereas when/how it declines.

```

Resizing AlignSpace: 100% | 57 / 57 [00:01:00:00, 34.48Bit/s]
Resizing Benign: 100% | 46 / 46 [00:04:00:00, 34.03Bit/s]
└ Loading dataset with ImageFolder...
  └ Running selfClean...
    └ Running selfClean...
[rank0]:[U10]:16:04:30.94602347 ProcessGroupNCL.cpp:4561] [PG ID 0 PG GUID 0 Rank 0] using GPU 0 to perform barrier as devices used by this process are currently unknown. This can potentially cause a hang if this rank is going to GPU mapping is incorrect. Specify device_ids in barrier() to force use of a particular device, or call init_process_group() with a device_id.
[025-05-12 16:04:31.157] | INFO | [Running on cuda]
[025-05-12 16:04:31.157] | INFO | [203 images and 13 batches loaded: there are 203 train images and 13 batches with a batch size of 16.]
[025-05-12 16:04:32.109] | INFO | [Student and Teacher are built: they are both pretrai..._dino network.
[025-05-12 16:04:32.109] | INFO | [Pretrained Weights not found. Training from scratch.
Epoch: 0, Train loss: 8.158937, Train stud/teach acc: 0.0000: 100% | 0 / 10 [00:09:07, 71.75Bit/s]
[025-05-12 16:04:32.109] | INFO | [Saving checkpoint: /home/chrai649/Desktop/DPHl work/AI_agents/M3Builder/dataset/skinl_dataset/DINO-skin_cancer_DDI_skin_dataset/DINO-skin_cancer/checkpoints/checkpoint0
Epoch: 1, Train loss: 8.158937, Train stud/teach acc: 0.0000: 100% | 1 / 10 [00:10:19:01:31, 10.20Bit/s]
[025-05-12 16:04:32.109] | INFO | [Saving checkpoint: /home/chrai649/Desktop/DPHl work/AI_agents/M3Builder/dataset/skinl_dataset/DINO-skin_cancer_DDI_skin_dataset/DINO-skin_cancer/checkpoints/checkpoint1
Epoch: 2, Train loss: 8.229657, Train stud/teach acc: 0.0000: 100% | 2 / 10 [00:10:19:01:31, 10.20Bit/s]
[025-05-12 16:04:32.109] | INFO | [Saving checkpoint: /home/chrai649/Desktop/DPHl work/AI_agents/M3Builder/dataset/skinl_dataset/DINO-skin_cancer_DDI_skin_dataset/DINO-skin_cancer/checkpoints/checkpoint2
Epoch: 3, Train loss: 8.161844, Train stud/teach acc: 0.0000: 20% | 3 / 10 [00:10:29:01:19, 9.98Bit/s]
[025-05-12 16:05:01.456] | INFO | [Saving checkpoint: /home/chrai649/Desktop/DPHl work/AI_agents/M3Builder/dataset/skinl_dataset/DINO-skin_cancer_DDI_skin_dataset/DINO-skin_cancer/checkpoints/checkpoint3
Epoch: 4, Train loss: 8.158937, Train stud/teach acc: 0.0000: 30% | 4 / 10 [00:10:29:01:19, 9.98Bit/s]
[025-05-12 16:05:01.456] | INFO | [Saving checkpoint: /home/chrai649/Desktop/DPHl work/AI_agents/M3Builder/dataset/skinl_dataset/DINO-skin_cancer_DDI_skin_dataset/DINO-skin_cancer/checkpoints/checkpoint4
Epoch: 5, Train loss: 8.193067, Train stud/teach acc: 0.0000: 40% | 5 / 10 [00:10:48:00:59, 9.97Bit/s]
[025-05-12 16:05:21.396] | INFO | [Saving checkpoint: /home/chrai649/Desktop/DPHl work/AI_agents/M3Builder/dataset/skinl_dataset/DINO-skin_cancer_DDI_skin_dataset/DINO-skin_cancer/checkpoints/checkpoint5
Epoch: 6, Train loss: 8.098457, Train stud/teach acc: 0.0000: 50% | 6 / 10 [01:00:08:00:39, 9.94Bit/s]
[025-05-12 16:05:31.149] | INFO | [Saving checkpoint: /home/chrai649/Desktop/DPHl work/AI_agents/M3Builder/dataset/skinl_dataset/DINO-skin_cancer_DDI_skin_dataset/DINO-skin_cancer/checkpoints/checkpoint6
Epoch: 7, Train loss: 8.082257, Train stud/teach acc: 0.0000: 60% | 7 / 10 [01:00:19:00:29, 9.89Bit/s]
[025-05-12 16:05:41.188] | INFO | [Saving checkpoint: /home/chrai649/Desktop/DPHl work/AI_agents/M3Builder/dataset/skinl_dataset/DINO-skin_cancer_DDI_skin_dataset/DINO-skin_cancer/checkpoints/checkpoint7
Epoch: 8, Train loss: 8.048531, Train stud/teach acc: 0.0000: 70% | 8 / 10 [01:00:28:00:19, 9.88Bit/s]
[025-05-12 16:05:51.629] | INFO | [Saving checkpoint: /home/chrai649/Desktop/DPHl work/AI_agents/M3Builder/dataset/skinl_dataset/DINO-skin_cancer_DDI_skin_dataset/DINO-skin_cancer/checkpoints/checkpoint8
Epoch: 9, Train loss: 8.009108, Train stud/teach acc: 0.0000: 80% | 9 / 10 [01:08:00:20, 10.07Bit/s]
[025-05-12 16:06:01.400] | INFO | [Saving checkpoint: /home/chrai649/Desktop/DPHl work/AI_agents/M3Builder/dataset/skinl_dataset/DINO-skin_cancer_DDI_skin_dataset/DINO-skin_cancer/checkpoints/checkpoint9
Epoch: 10, Train loss: 8.089119, Train stud/teach acc: 0.0000: 90% | 10 / 10 [01:39:08:00, 10.08Bit/s]
[025-05-12 16:06:01.374] | INFO | [Saving current best: model_best.pth... | 15 / 15 [00:01:00:00, 10.07Bit/s]
[025-05-12 16:06:12.317] | INFO | [Saving current best: model_best.pth... | 10 / 10 [01:40:00:00, 10.02Bit/s]
[025-05-12 16:06:12.317] | INFO | [Fitting cleaner on representation space: (203, 192) | 15 / 15 [00:01:00:00, 10.07Bit/s]
Creating Dataset Reprojection: 100% | Processing possibly new duplicates: 100% | 3 / 3 [00:00:00:00, 49.23Bit/s]
Processing possibly new duplicates: 100% | Processing possibly off-topic samples: 40516 [00:00: 10.63GB, 0.01Bit/s]
Processing possibly off-topic samples: 40516 [00:00: 10.63GB, 0.01Bit/s]
[025-05-12 16:06:14.053] | INFO | [Returning as data frame requires extensive memory.
[025-05-12 16:06:14.053] | INFO | [Returning as data frame requires extensive memory.

```

Figure 35: Data-cleaning by learning the representation space of DDI skin cancer dataset using DINO

```

6480
6481
6482
6483
6484
6485
6486
6487
6488
6489
6490
6491
6492
6493
6494
6495
6496
6497
6498
6499
6500
6501
6502
6503
6504
6505
6506
6507
6508
6509
6510
6511
6512
6513
6514
6515
6516
6517
6518
6519
6520
6521
6522
6523
6524
6525
6526
6527
6528
6529
6530
6531
6532
6533

Resting Keratosis: 100%
Resting Carcinoma: 100%
Resting Allergic: 100%
Resting Basalcell: 100%
Resting Eczema: 100%
Resting acne: 100%
Running dataset with ImageFolder...
Running Selfclean...
Selfclean loaded
[rank0]:[K512 16:01:08.952787200 ProcessGroupNCCL.cpp:1641] [BG ID 0 MC GPU 0 Rank 0] using GPU 0 to perform barrier as devices used by this process are currently unknown. This can potentially cause a
hang if the GPU to GPU mapping is incorrect. Specify device_ids in barrier() to force use of a particular device, or call init_process_group() with a device_id.
1925-05-12 10:01:09.121 | INFO | Data loaded: there are 180 train images and 12 batches with a batch size of 16.
Using locally downloaded DINO checkpoint.
Model Agent and Teacher are built: they are both pretrained_imagenet_dino network.
1925-05-12 10:01:09.076 | INFO | Found checkpoint at /home/chr16419/Desktop/DPHill/work/AI_agents/M3Builder/dataset/skin_dataset/DINO-skin_cancer_augmented_skin_condition_dataset_kaggle/DINO-skin_cancer/checkpoints/model_best.pth
Creating dataset representation: 100%
1925-05-12 10:01:11.580 | INFO | Fitting cleaner on representation space: (100, 192)
Processing possible near duplicates: 100%
Processing possible off-topic samples: 3591t [0:0:0 10312.081t/s]
1925-05-12 10:01:11.580 | WARNING | Returning as dataframe requires extensive memory.
1925-05-12 10:01:11.580 | WARNING | Returning as dataframe requires extensive memory.
1925-05-12 10:01:11.580 | WARNING | Returning as dataframe requires extensive memory.
Saved CSVs to /home/chr16419/Desktop/DPHill/work/AI_agents/M3Builder/dataset/skin_dataset_cleaned/augmented_skin_condition_dataset_kaggle
Datacleaner_Content Data Cleaning Complete

```

Figure 36: Data-cleaning by learning the representation space of augmented-skin-condition-dataset-kaggle using DINO