Agent Role Structure and Operating Characteristics in Large Language Model Clinical Classification: A Comparative Study of Specialist and Deliberative Multi-Agent Protocols

Callum Anderson[a]

[a] *Faculty of Engineering, University of Ottawa, 800 King Edward Avenue, Ottawa, Ontario K1N 6N5, Canada*

Abstract

Large language models (LLMs) are increasingly deployed in structured clinical decision support, yet the architectural effects of internal role decomposition within multi-agent systems remain poorly isolated. Prior comparisons of single-agent and multi-agent prompting frequently confound workflow structure with changes in model configuration, training, or decoding. We present a controlled architectural study of role-structured inference under fixed model parameters, isolating internal role decomposition as the sole manipulated variable. Two deterministic multi-agent protocols, Generic Deliberative (GD) and Feature-Specialist (FS), are evaluated under identical base weights, decoding settings, computational budget, and adjudication logic. Across two tabular clinical benchmarks (UCI Cleveland Heart Disease and Pima Indians Diabetes), altering role structure alone systematically reshapes operating characteristics. On Cleveland, FS improves accuracy by 0.07 and macro-F1 by 0.06 relative to GD, while shifting the operating point toward higher specificity (+0.22) and lower sensitivity (-0.13), substantially reducing false positives. On Pima, architectural effects reverse direction: GD achieves the strongest macro performance (accuracy 0.68, macro-F1 0.64), whereas FS induces pronounced class asymmetry (recall 0.95 for the positive class and 0.27 for the negative class). These findings demonstrate that internal role decomposition functions as a structured inductive bias that can materially alter error distributions without modifying model parameters. Multi-agent prompt architecture should therefore be treated as an explicit mechanism for controlling sensitivity-specificity trade-offs in safety-sensitive LLM decision systems.

Keywords: Large language models, Multi-agent systems, Role decomposition, Structured prompting, Clinical decision support, Operating characteristics

1. Introduction

Recent advances in large language models (LLMs) have increased interest in their use in clinical settings. Healthcare institutions now maintain extensive structured electronic health record (EHR) repositories, and artificial intelligence techniques are increasingly being applied to these data for clinical prediction, decision support, and workflow optimization [1]. Reviews of medical natural language processing in the LLM era show that modern models encode substantial biomedical knowledge and can support clinically relevant reasoning, particularly when adapted through instruction tuning and task-specific guidance [2]. Empirical studies further demonstrate that LLMs can process structured clinical artifacts, such as eligibility criteria and patient descriptors, and produce representations that support grouping and classification tasks [3]. LLM-based systems have also been explored for identifying social and contextual patient information from clinical records and incorporating these factors into predictive workflows [4]. Together, this body of work suggests that LLMs can operate over structured clinical information and support medical classification-oriented tasks. In medical contexts, locally deployable large language models are of particular interest, as they allow sensitive patient data to remain within institutional infrastructure while maintaining operational autonomy and regulatory compliance [5]. However, despite increasing adoption of multi-agent prompting workflows, the specific effects of internal role decomposition within LLM-based decision systems remain poorly isolated, and it is unclear whether prompt-level agent specialization alone can systematically influence classification behavior in clinical prediction tasks.

Efforts to improve reliability and reasoning depth in LLM-based systems have increasingly adopted multi-agent or chained architectures. Rather than relying on a single model invocation, these systems decompose complex tasks into coordinated sub-tasks executed by multiple LLM in-

*Corresponding author
Email address: cande007@uottawa.ca (Callum Anderson)

stances, often with explicit roles or sequential dependency structures [6, 7]. These approaches draw conceptual parallels to ensemble-based decision systems, where multiple decision-makers contribute complementary perspectives to reduce error and improve robustness [8]. In computational settings, structured LLM workflows have been shown to make intermediate reasoning steps explicit and to introduce validation and adjudication mechanisms between agents, thereby increasing the auditability of system behavior and allowing more deliberate control over task execution [6]. These architectural patterns are particularly relevant for safety-sensitive domains such as clinical and diagnostic decision support, where structured multi-step reasoning and cross-checking may offer advantages over single-agent architectures. In these environments, shaping operating characteristics, including tradeoffs among sensitivity, specificity, and precision, is critical, as clinical workflows often prioritize minimizing missed diagnoses even at the expense of increased false positives [9].

Although a growing number of multi-agent LLM frameworks introduce layered, cooperative, or dynamically orchestrated agent structures, most prior research has emphasized architectural design and application-level performance rather than tightly controlled analysis of internal role decomposition [10, 11, 6]. Relatively little attention has been given to isolating prompt-level role structure as an independent architectural variable under controlled experimental conditions. Therefore, it remains unclear how much internal agent specialization or overlapping full-view reasoning alone contributes to observed differences in classification behavior and operating characteristics.

To address this specific gap, we conduct a controlled comparison of two structured multi-agent architectures that differ only in internal role decomposition and patient data access. Both are implemented as deterministic Directed Acyclic Graph (DAG) workflows operating on two tabular clinical benchmarks, the UCI Cleveland heart disease dataset [12] and the Pima Indians Diabetes dataset [13], with agents running in parallel prior to a downstream adjudication step. In the GD configuration, generalist agents independently evaluate the complete feature set, whereas in the FS configuration, agents are assigned distinct clinically relevant feature domains and produce complementary partial assessments. Intermediate evidence representations are standardized across architectures, and model weights, decoding temperature, computational budget, aggregation logic, and implementation code are held constant. This design isolates role structure and information routing, enabling direct assessment of how internal agent organization alone shifts classification performance and redistributes sensitivity–specificity trade–offs across distinct clinical prediction tasks.

2. Methods

2.1. Datasets

We evaluate both architectures on two structured clinical benchmarks originally released through the UCI Machine Learning Repository: the Cleveland Heart Disease dataset [12] and the Pima Indians Diabetes dataset [13]. Both datasets are widely used for binary diagnostic classification and contain tabular patient-level clinical measurements.

The processed Cleveland subset contains 303 patient records with 13 structured clinical features spanning demographic variables, exercise test results, and imaging-derived indicators. The positive class prevalence in the binarized Cleveland dataset is 0.459 (45.9%).

The Pima Indians Diabetes dataset contains 768 patient records with 8 structured clinical features including laboratory measurements, anthropometric indicators, and family history scores. The positive class prevalence in the Pima dataset is 0.349 (34.9%).

For consistency across experiments, both datasets use a unified binary target variable `disease_present`, where 0 denotes absence of disease and 1 denotes disease presence. Although the semantic interpretation differs by dataset, representing cardiovascular disease in Cleveland and diabetes in Pima, the encoding is identical, enabling controlled comparison of architectural effects across tasks.

2.1.1. Preprocessing and Labeling

For the Cleveland dataset, the original target variable (`num`), which encodes heart disease severity on an ordinal scale from 0 to 4, was binarized. Records with a value of 0 were mapped to `disease_present` value 0 (no disease), whereas records with values 1 through 4 were mapped to value 1 (disease present).

The Pima dataset natively provides a binary outcome variable (`Outcome`), which was directly mapped to the unified `disease_present` target without modification.

For the Cleveland dataset, missing values were handled via per-feature mode imputation computed across the processed subset. For each variable containing missing observations, the most frequent observed value was used to replace null entries.

The Pima dataset contains zero-coded entries in certain physiological variables that are commonly treated as missing in distributed versions of the dataset. In this study, all features were preserved in their original encoded form and no additional missing-value indicators were introduced into the serialized prompts presented to the language model.

Each dataset was serialized into a structured textual representation for LLM input. Cleveland features include `age_years`, `sex`, `chest_pain_type`, `resting_blood_pressure_mmHg`, `serum_cholesterol_mg_dL`, `fasting_blood_sugar_over_120_mg_dL`, `resting_ecg_result`,

max_heart_rate_bpm, exercise_induced_angina, st_depression_exercise_relative_to_rest, st_segment_slope_peak_exercise, num_major_vessels_fluoroscopy, and thalassemia_test_result.

For the Pima Indians Diabetes dataset, no feature transformations were applied beyond renaming columns to standardized LLM-facing identifiers while preserving all original numeric values. Pima features include pregnancies, plasma_glucose_mg_dL, diastolic_blood_pressure_mmHg, triceps_skinfold_thickness_mm, serum_insulin_uU_mL, body_mass_index, diabetes_pedigree_function, and age_years. A complete mapping of raw variables, encoded values, and LLM-facing feature names for both datasets is provided in Appendix Appendix B.

2.2. Prompting Protocols

We implement two structured multi-agent protocols that share identical model configuration, decoding parameters, adjudication logic, and a common structured evidence representation consisting of signal, strength, and justification fields. Both protocols operate as DAG workflows in which upstream agents generate structured intermediate assessments that are consumed by a downstream adjudicator. The protocols differ only in internal role decomposition and information routing.

2.2.1. Protocol A: Feature-Specialist Architecture (FS)

The Feature-Specialist (FS) architecture comprises two parallel specialist agents, each restricted to a single clinically defined feature from the patient record. Each specialist produces a structured opinion describing the evidentiary contribution of its assigned feature, including (1) a directional signal (supports_disease, supports_no_disease, or unclear), (2) a categorical strength rating (low, medium, high), and (3) a brief justification note explicitly referencing the feature value. Specialists operate independently and are prohibited from reasoning over features outside their assigned scope.

The two selected features in each dataset were chosen to represent distinct and complementary information domains within the patient record. For each benchmark, one feature from each of two different clinical or physiological categories was assigned to a separate specialist agent. This ensured that specialists operated on non-overlapping subsets of the record while preserving structural comparability across tasks. The design isolates the effect of role decomposition and information routing within each dataset.

Formally, given a serialized patient record x, each specialist produces an opinion

$$o_i = (\phi_i,\ \text{signal}_i,\ \text{strength}_i,\ \text{note}_i), \quad i \in \{1, 2\}, \tag{1}$$

where ϕ_i denotes the feature assigned to specialist i. The adjudicator then receives the full record and both opinions and outputs the final binary prediction:

$$\hat{y} = f_\theta(x, o_1, o_2), \quad \hat{y} \in \{\texttt{disease_present}, \texttt{no_disease}\}, \tag{2}$$

where f_θ denotes a third LLM call under fixed model weights and decoding settings. The adjudicator is instructed to treat strength as ordinal evidence (high > medium > low); when specialist evidence conflicts or is weak or unclear, it may consult the full record to decide. If genuinely uncertain, no_disease is used as a conservative tie-break. If any agent fails to produce valid JSON, a conservative fallback output is substituted. All instances of fallback utilization are transparently reported in the Results section and fully logged in the public Git repository accompanying this study.

As shown in Figure 1, the FS protocol assigns each upstream agent a single feature and routes their structured outputs to an adjudicator.

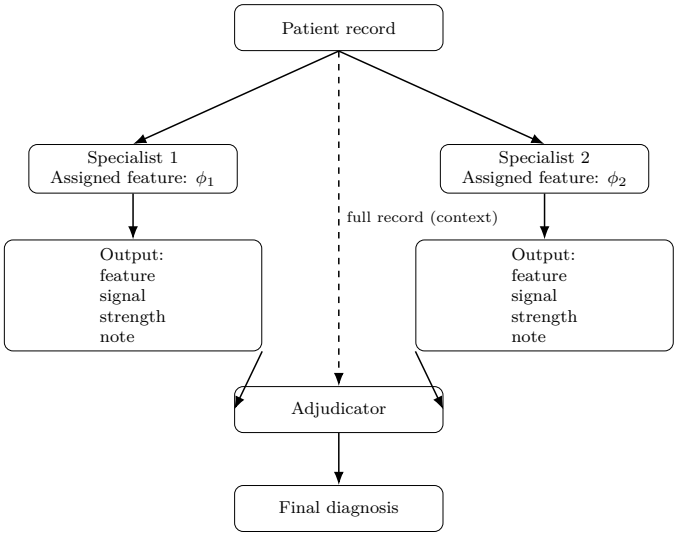

Figure 1: **Feature-specialist (FS) architecture.** Two agents independently evaluate distinct assigned features (ϕ_1, ϕ_2) from the patient record. Their structured outputs (feature, signal, strength, note) are routed to an adjudicator, which also receives the full record and produces the final binary diagnosis. The specific feature assignments are dataset-dependent.

2.2.2. Protocol B: Generic Deliberative Architecture (GD)

The Generic Deliberative (GD) architecture comprises two parallel generalist clinician agents, each evaluating the full patient record. Each clinician produces a structured opinion consisting of (1) a directional signal (supports_disease, supports_no_disease, or unclear), (2) a categorical strength rating (low, medium, high), and (3) a brief justification citing one to two key record-level factors. The clinicians operate independently without inter-agent communication.

Formally, given a serialized patient record x, each clinician produces an opinion

$$o_i = (\texttt{scope}_i,\ \texttt{signal}_i,\ \texttt{strength}_i,\ \texttt{note}_i),\quad i \in \{1, 2\}, \tag{3}$$

where $\texttt{scope}_i = \texttt{overall_record}$ for both clinicians, indicating that each agent evaluates the complete patient record rather than a restricted feature subset. The adjudicator then receives the full record and both opinions and outputs the final binary prediction:

$$\hat{y} = f_\theta(x, o_1, o_2),\quad \hat{y} \in \{\texttt{disease_present}, \texttt{no_disease}\}, \tag{4}$$

where f_θ denotes a third LLM call under fixed model weights and decoding settings. The adjudicator is instructed to treat strength as ordinal evidence (high > medium > low); when clinician evidence conflicts or is weak or unclear, it may consult the full record to decide. If genuinely uncertain, `no_disease` is used as a conservative tie-break, and the adjudicator is explicitly instructed not to systematically favor either clinician. If any agent fails to produce valid JSON, a conservative fallback output is substituted. All instances of fallback utilization are transparently reported in the Results section and fully logged in the public Git repository accompanying this study.

Figure 2 shows the GD protocol, in which both upstream agents evaluate the full record and provide independent opinions to the adjudicator.

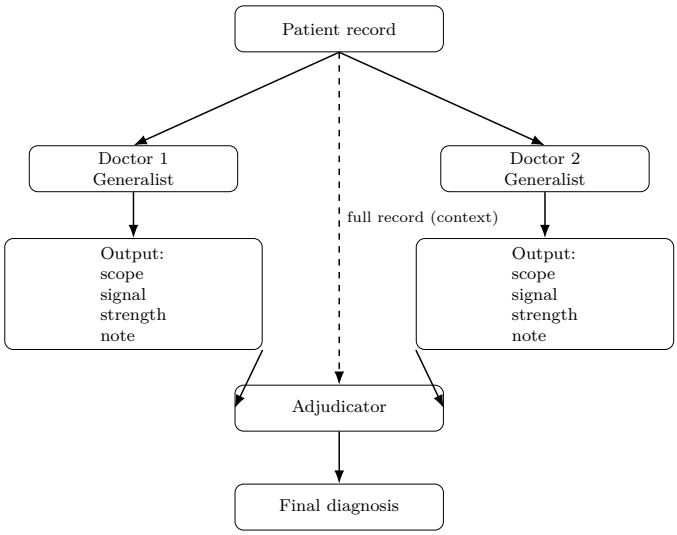

Figure 2: **Generic deliberative (GD) architecture.** Two generalist clinician agents independently evaluate the full patient record and produce structured outputs (scope, signal, strength, note), where `scope` is set to `overall_record`. These outputs are routed to an adjudicator, which also receives the full record and produces the final binary diagnosis.

2.3. Implementation and Reproducibility

All experiments were implemented in Python 3.13, with multi-agent orchestration handled via LangChain and local inference performed through Ollama. All architectures used identical base model weights and deterministic decoding settings to ensure that internal role structure was the sole manipulated variable. The underlying model was `llama3.1:8b`, executed locally with temperature set to zero. Structured outputs were enforced through strict JSON schema constraints and programmatically validated prior to adjudication. Core libraries included `ollama` for inference, `langchain` for workflow orchestration and prompt templating, and `pandas` and `numpy` for preprocessing and evaluation.

All source code, prompt templates, and evaluation scripts are publicly available in the associated GitHub repository:

`https://github.com/CallumGA/llm-multi-agent-role-decomposition`

The datasets used in this study are publicly available:

- UCI Heart Disease dataset (Cleveland subset): `https://archive.ics.uci.edu/dataset/45/heart+disease`
- Pima Indians Diabetes dataset (UCI-origin, accessed via Kaggle): `https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database`

2.4. Evaluation Metrics

For all evaluated systems, we report standard classification performance metrics, including overall Accuracy as well as macro-averaged Precision, Recall, and F1-score. Confusion matrices are provided for both the Generic Deliberative (GD) and Feature-Specialist (FS) architectures to give a transparent breakdown of class-wise predictions. To characterize operating-point behavior, we explicitly report sensitivity–specificity trade-offs, where sensitivity is defined as recall for the `disease_present` class and specificity as recall for the `no_disease` class.

3. Results

3.1. Overall Performance

Table 1 summarizes overall macro performance across the three protocols.

3.2. Class-wise Performance

Table 2 reports class-wise precision, recall, and F1-score across datasets and protocols.

3.3. Confusion Matrices

Table 3 reports confusion matrix counts (rows = true class, columns = predicted class) for the multi-agent protocols across datasets.

Table 1: Overall performance across evaluation datasets.

System	Acc.	Prec. (M)	Rec. (M)	F1 (M)
Cleveland Dataset				
Single-Agent Protocol	0.69	0.71	0.70	0.69
Generic Deliberative (GD)	0.65	0.66	0.66	0.65
Feature-Specialist (FS)	0.72	0.72	0.71	0.71
Pima Indians Diabetes Dataset				
Single-Agent Protocol	0.64	0.63	0.64	0.63
Generic Deliberative (GD)	0.68	0.65	0.64	0.64
Feature-Specialist (FS)	0.51	0.66	0.61	0.49

Table 2: Class-wise performance across evaluation datasets.

System	Class	Prec.	Rec.	F1
Cleveland Dataset				
Single-Agent	no_disease	0.80	0.57	0.66
	disease_present	0.62	0.83	0.71
Generic Deliberative (GD)	no_disease	0.72	0.60	0.65
	disease_present	0.60	0.72	0.66
Feature-Specialist (FS)	no_disease	0.70	0.82	0.76
	disease_present	0.74	0.59	0.66
Pima Indians Diabetes Dataset				
Single-Agent	no_disease	0.76	0.66	0.71
	disease_present	0.49	0.61	0.54
Generic Deliberative (GD)	no_disease	0.74	0.79	0.77
	disease_present	0.55	0.48	0.51
Feature-Specialist (FS)	no_disease	0.91	0.27	0.42
	disease_present	0.41	0.95	0.57

Table 3: Confusion matrices (rows = true class, columns = predicted class) across evaluation datasets.

System	TN	FP	FN	TP
Cleveland Dataset				
Generic Deliberative (GD)	98	66	39	100
Feature-Specialist (FS)	135	29	57	82
Pima Indians Diabetes Dataset				
Generic Deliberative (GD)	396	104	139	129
Feature-Specialist (FS)	135	365	14	254

4. Discussion

4.1. Interpretation of Principal Findings

The performance differences between the Feature-Specialist (FS) and Generic Deliberative (GD) protocols indicate that internal role structure influences how clinical evidence is combined during inference. On the Cleveland dataset, FS achieves higher overall accuracy (0.72 vs. 0.65) and macro-F1 (0.71 vs. 0.65), but these improvements reflect a redistribution of errors rather than uniform gains. As shown in Table 3, FS substantially increases true negatives (135 vs. 98) and reduces false positives (29 vs. 66), indicating improved identification of patients without disease. This pattern is accompanied by fewer true positives (82 vs. 100) and more false negatives (57 vs. 39), reflecting lower sensitivity and a shift toward greater specificity. One explanation is that separating feature evaluation constrains how positive evidence accumulates: when specialists reason over restricted inputs, weak or ambiguous signals may be less likely to combine into an overall positive classification. In contrast, the GD protocol evaluates features jointly within a single reasoning trajectory, allowing multiple modest indicators to interact and collectively influence the prediction.

On the Pima Indians Diabetes dataset, structural effects again emerge, but manifest differently. Here, GD achieves the strongest overall macro performance (accuracy 0.68, macro-F1 0.64), whereas FS produces highly asymmetric class behavior. The FS protocol yields very high recall for the positive class (0.95) but very low recall for the negative class (0.27), corresponding to a large increase in false positives (365 vs. 104) relative to GD. Thus, in this setting, role decomposition shifts the operating point toward high sensitivity and reduced specificity, the opposite direction observed on Cleveland. The Pima dataset consists primarily of continuous numerical features, which may influence how evidence is partitioned and aggregated across agents. However, the central finding concerns within-dataset comparisons: in both benchmarks, holding model weights, decoding settings, and computational budget fixed, altering only internal role structure produces systematic changes in sensitivity–specificity trade-offs.

While it is expected that prompt variations can influence LLM outputs, the magnitude and systematic nature of the operating-point shifts observed here under fully deterministic conditions demonstrate that internal role decomposition alone can materially alter classification behavior. These differences arise despite identical base model weights and inference settings, indicating that role organization functions as a structured inductive bias rather than a superficial prompting variation.

In clinical contexts, such trade-offs are operational rather than purely statistical. Reductions in false positives may decrease unnecessary follow-up testing or patient

anxiety, whereas increases in false negatives may delay diagnosis or treatment initiation. The appropriate balance depends on intended use, such as screening versus confirmatory decision support. Overall, these findings demonstrate that multi-agent role structure does not simply improve or degrade performance globally; instead, it systematically reshapes the error profile. Careful evaluation of this redistribution is therefore essential when deploying structured LLM systems in safety-sensitive decision environments [14].

4.2. Relationship to Prior Work

The observed behavioral differences align with modular design principles in which inputs are routed through distinct components whose outputs are later combined. In sparsely activated mixture-of-experts models, routing determines which submodules contribute to a prediction and thus can change the effective input–output mapping without any weight updates [15]. This supports the broader view that internal organization and information routing can shape decision behavior even when model parameters are fixed.

Evidence from multi-agent prompting indicates that explicit role decomposition can shift performance relative to single-agent baselines. Multi-agent simplification workflows have been shown to outperform single-agent approaches on the same dataset, with gains attributed to staged role partitioning and validation rather than changes to the underlying model [16]. In multi-agent reinforcement learning, architectural asymmetry has likewise been shown to influence outcomes, including the introduction of specialized agents within value-decomposition frameworks in heterogeneous environments [17].

Workflow-level analyses of LLM chaining further emphasize that topology affects controllability, transparency, and output characteristics [6]. However, many prior comparisons confound architectural structure with concurrent changes in prompts, model configuration, or training. In contrast, the present study isolates internal role structure as the sole manipulated factor under fixed weights and deterministic decoding, enabling clearer attribution of operating-characteristic shifts to architectural organization.

4.3. Limitations

This study evaluates role-structured prompting on two publicly available benchmark datasets (UCI Cleveland Heart Disease and Pima Indians Diabetes) under binary classification, limiting generalizability to other populations, multiclass tasks, and higher-dimensional EHR settings. Both datasets are relatively small and curated, and may not reflect real-world clinical noise, missingness patterns, or distributional shift.

Given the public availability of these datasets, prior exposure during pretraining of the underlying model (`llama3.1:8b`) cannot be excluded; however, identical base weights were used across all agents, so any such exposure would affect both protocols equally. Experiments were conducted using an 8B-parameter model to enable local execution within realistic institutional resource constraints. Larger models may exhibit different operating characteristics, and future work should examine whether the observed architectural effects persist across scales. All experiments used deterministic decoding with temperature set to zero and a limited set of agent configurations with fixed aggregation logic. No calibration analysis, probabilistic decoding, or external validation on independent cohorts was performed.

5. Conclusions and Future Work

This study presented a controlled comparison of two deterministic multi-agent architectures for binary clinical classification across the UCI Cleveland Heart Disease and Pima Indians Diabetes datasets, differing only in internal role decomposition and information routing under fixed model and inference settings. Across both datasets, altering role structure produced systematic shifts in operating characteristics within each task. On Cleveland, the Feature-Specialist protocol increased overall accuracy and macro-F1 relative to the Generic Deliberative protocol but shifted the operating point toward greater specificity and reduced sensitivity. On Pima, role decomposition again materially altered class-wise behavior, producing a pronounced redistribution of sensitivity and specificity in the opposite direction. These findings demonstrate that internal agent organization can reshape error profiles without modifying model parameters, functioning as a structured inductive bias in LLM-based clinical classifiers. Prompt-level architectural design should therefore be treated as a core modeling decision rather than a superficial implementation detail.

Future work should validate these effects across additional cohorts and structured clinical benchmarks with external testing, extend the comparison to multiclass and higher-dimensional EHR tasks, and examine calibration and probabilistic decoding behavior. Agent-count and role-ablation studies may clarify how architectural granularity influences operating characteristics. Importantly, future research should seek to formalize a structured framework for tuning LLM behavior through systematic manipulation of role decomposition and information routing, enabling more principled control over sensitivity–specificity trade-offs in safety-sensitive applications. Evaluation within simulated clinical workflows may further clarify how architecture-induced error redistributions affect deployment-relevant decisions.

Appendices

Appendix A. Full Prompt Specifications

This appendix provides the exact zero-shot prompt specifications, patient record rendering format, and fallback behavior used for all experimental conditions (single-agent, generic multi-agent, and specialist multi-agent). Prompts are reproduced verbatim to ensure reproducibility.

Appendix A.1. Patient Record Serialization

All models received patient records in a structured text format. The serialization template differed by dataset but preserved a consistent human-readable layout across experiments.

UCI Cleveland Heart Disease Dataset

```
Patient information:
- Age: {age_years}
- Sex: {sex}
- Chest pain type: {chest_pain_type}
- Resting blood pressure:
    {resting_blood_pressure_mmHg} mmHg
- Serum cholesterol: {serum_cholesterol_mg_dL} mg/dL
- Fasting blood sugar > 120 mg/dL:
    {fasting_blood_sugar_over_120_mg_dL}
- Resting ECG result: {resting_ecg_result}
- Maximum heart rate achieved: {max_heart_rate_bpm}
    bpm
- Exercise-induced angina: {exercise_induced_angina}
- ST depression induced by exercise:
    {st_depression_exercise_relative_to_rest}
- ST segment slope during peak exercise:
    {st_segment_slope_peak_exercise}
- Number of major vessels (fluoroscopy):
    {num_major_vessels_fluoroscopy}
- Thalassemia test result: {thalassemia_test_result}
```

Pima Indians Diabetes Dataset

```
Patient information:
- Age: {age_years}
- Pregnancies: {pregnancies}
- Plasma glucose concentration (2-hour OGTT):
    {plasma_glucose_mg_dL} mg/dL
- Diastolic blood pressure:
    {diastolic_blood_pressure_mmHg} mmHg
- Triceps skinfold thickness:
    {triceps_skinfold_thickness_mm} mm
- 2-hour serum insulin: {serum_insulin_uU_mL} uU/mL
- Body mass index (BMI): {body_mass_index} kg/m^2
- Diabetes pedigree function:
    {diabetes_pedigree_function}
```

Appendix A.2. Single-Agent Baseline Prompt

```
You are a clinician performing a binary diagnostic
    classification task.

You will receive the full patient record.

Your job:
Return the final diagnosis:
- "disease_present" or "no_disease"

Rules (do not mention these in output):
- Consider all relevant factors in the patient record.
- If evidence clearly supports disease, return
    "disease_present".
- Otherwise return "no_disease".
- If genuinely uncertain, choose "no_disease".

Output MUST be a single JSON object and nothing else.
{ "diagnosis": "disease_present" OR "no_disease" }
```

Appendix A.3. Generic Deliberative (GD) Protocol

Appendix A.3.1. Doctor Opinion Prompt

```
You are a clinician reviewing the FULL patient record.

Task:
Provide an overall opinion about whether the FULL
    RECORD supports:
- supports_disease
- supports_no_disease
- unclear

Guidelines (do not mention these in output):
- Use "supports_disease" only if overall evidence
    clearly supports disease.
- Use "supports_no_disease" only if overall evidence
    clearly supports no disease.
- Otherwise use "unclear".
- "strength" reflects how strong the overall evidence
    is (high > medium > low).
- "note" must be ONE short sentence citing 1 to 2 key
    factors from the record (no long reasoning).

Output MUST be a single JSON object and nothing else:
{
  "scope": "overall_record",
  "signal": "supports_disease" OR
    "supports_no_disease" OR "unclear",
  "strength": "low" OR "medium" OR "high",
  "note": "one short sentence"
}
```

Appendix A.3.2. Adjudicator Prompt

```
You are an adjudicator for a binary clinical
    classification task.

You will receive:
1) The full patient record
```

```
2) Two doctor opinions about the full record (signal
     + strength)

Your job:
Return the final diagnosis:
- "disease_present" or "no_disease"

Rules (do not mention these in output):
- Treat each doctor opinion as evidence about the
     overall record; it may be low quality or wrong.
- Weigh evidence by "strength" (high > medium > low).
- If both doctors strongly support the same
     direction, follow that direction.
- If evidence conflicts or is weak/unclear, use the
     full patient record to decide.
- If still genuinely uncertain after considering the
     full record, choose "no_disease" as a tie-break.
- Do not automatically prefer Doctor 1 or Doctor 2.

Output MUST be a single JSON object and nothing else.
{ "diagnosis": "disease_present" OR "no_disease" }
```

Appendix A.4. Feature-Specialist (FS) Protocol

Appendix A.4.1. Feature Specialist Prompt

```
You are a clinical feature specialist. You ONLY
     evaluate the feature: "{feature_name}".

Task:
Given the patient record, decide whether THIS FEATURE
     ALONE:
- supports_disease
- supports_no_disease
- unclear

Guidelines (do not mention these in output):
- Use "supports_disease" only if this feature value
     is a meaningful risk indicator by itself.
- Use "supports_no_disease" only if this feature
     value is meaningfully reassuring by itself.
- Otherwise use "unclear".
- "strength" reflects how strongly this single
     feature points in that direction (not overall
     diagnosis):
  - high: strong / clear signal
  - medium: moderate signal
  - low: weak, borderline, missing, or noisy
- "note" must be ONE short sentence that explicitly
     cites the patients "{feature_name}" value and why
     it points that way.
- Do NOT use any other features in your reasoning.

Output MUST be a single JSON object and nothing else,
     with EXACTLY these keys:
{
  "feature": "{feature_name}",
  "signal": "supports_disease" OR
     "supports_no_disease" OR "unclear",
  "strength": "low" OR "medium" OR "high",
  "note": "one short sentence referencing the
     patient's {feature_name} value"
}
```

Appendix A.4.2. Specialist Adjudicator Prompt

```
You are an adjudicator for a binary clinical
     classification task.

You will receive:
1) The full patient record
2) Two specialist opinions (each about ONE feature
     only)

Your job:
Return the final diagnosis:
- "disease_present" or "no_disease"

Rules (do not mention these in output):
- Treat each specialist opinion as evidence about a
     single feature; it may be low quality or wrong.
- Weigh evidence by "strength" (high > medium > low).
- If both specialists strongly support the same
     direction, follow that direction.
- If evidence conflicts or is weak/unclear, use the
     full patient record to decide.
- If still genuinely uncertain after considering the
     full patient record, choose "no_disease" as a
     tie-break.

Output MUST be a single JSON object and nothing else.
{ "diagnosis": "disease_present" OR "no_disease" }
```

Appendix B. Full Feature Mapping

This study uses two publicly available datasets from the UCI Machine Learning Repository: the UCI Heart Disease dataset (Cleveland subset) [12] and the Pima Indians Diabetes dataset [13]. Feature definitions and categorical encodings follow the official dataset documentation distributed with each download.

UCI Heart Disease (Processed Cleveland Subset)

Table B.1 lists the full feature schema and encodings used in this study.

Pima Indians Diabetes Dataset

Table B.2 lists the full feature schema and encodings used for the Pima dataset.

Funding

No external funding was received for this work.

Conflicts of Interest

The author declares no competing interests.

Ethics Statement

This study uses publicly available, fully de-identified datasets: the UCI Heart Disease (Cleveland subset) dataset and the Pima Indians Diabetes dataset. No new data were collected and no human participants were recruited. As the analysis relied exclusively on secondary use of anonymized public data, institutional review board approval was not required.

Data and Code Availability

The datasets used in this study are publicly available:

UCI Heart Disease dataset (Cleveland subset): https://archive.ics.uci.edu/dataset/45/heart+disease

Pima Indians Diabetes dataset (Kaggle version): https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

All source code, prompt templates, and evaluation scripts are publicly available at: https://github.com/CallumGA/llm-multi-agent-role-decomposition.

Table B.1: Full feature schema and categorical encodings for the UCI Heart Disease (Processed Cleveland subset).

# Feature Name	Orig. Index	Raw Name	Description	Encoded Values / Units	LLM-Friendly
1	#3	age	Age of the patient	Years	`age_years`
2	#4	sex	Biological sex	1 = male; 0 = female	`sex`
3	#9	cp	Chest pain type	1 = typical angina 2 = atypical angina 3 = non-anginal pain 4 = asymptomatic	`chest_pain_type`
4	#10	trestbps	Resting blood pressure	mm Hg	`resting_blood_pressure_mmHg`
5	#12	chol	Serum cholesterol	mg/dL	`serum_cholesterol_mg_dL`
6	#16	fbs	Fasting blood sugar > 120 mg/dL	1 = true; 0 = false	`fasting_blood_sugar_over_120_mg_dL`
7	#19	restecg	Resting electrocardiographic results	0 = normal 1 = ST-T wave abnormality 2 = left ventricular hypertrophy	`resting_ecg_result`
8	#32	thalach	Maximum heart rate achieved	Beats per minute	`max_heart_rate_bpm`
9	#38	exang	Exercise-induced angina	1 = yes; 0 = no	`exercise_induced_angina`
10	#40	oldpeak	ST depression induced by exercise relative to rest	Numeric (float)	`st_depression_exercise_relative_to_rest`
11	#41	slope	Slope of peak exercise ST segment	1 = upsloping 2 = flat 3 = downsloping	`st_segment_slope_peak_exercise`
12	#44	ca	Number of major vessels colored by fluoroscopy	Integer (0–3)	`num_major_vessels_fluoroscopy`
13	#51	thal	Thalassemia test result	3 = normal 6 = fixed defect 7 = reversible defect	`thalassemia_test_result`
14	#58	num	Diagnosis of heart disease (target variable)	0 = no disease 1–4 = disease present (increasing severity)	`disease_present` (binary derived)

Table B.2: Full feature schema and encodings for the Pima Indians Diabetes dataset.

# Feature Name	Raw Name	Description	Units / Encoding	Notes	LLM-Friendly
1	pregnancies	Number of pregnancies	Integer	–	`pregnancies`
2	glucose	Plasma glucose concentration (2-hour OGTT)	mg/dL	Continuous	`plasma_glucose_mg_dL`
3	bloodpressure	Diastolic blood pressure	mm Hg	Continuous	`diastolic_blood_pressure_mmHg`
4	skinfold	Triceps skinfold thickness	mm	Continuous	`triceps_skinfold_thickness_mm`
5	insulin	2-hour serum insulin	μU/mL	Continuous	`serum_insulin_uU_mL`
6	bmi	Body mass index	kg/m^2	Continuous	`body_mass_index`
7	pedigree	Diabetes pedigree function	Numeric (float)	Continuous	`diabetes_pedigree_function`
8	age	Age of the patient	Years	Continuous	`age_years`
9	outcome	Diabetes diagnosis (target variable)	0 = no disease 1 = disease present	Binary	`disease_present`

References

[1] D. Fernandes Prabhu, V. Gurupur, A. Stone, E. Trader, Integrating artificial intelligence, electronic health records, and wearables for predictive, patient-centered decision support in healthcare, Healthcare 13 (21) (2025) 2753. doi:10.3390/healthcare13212753.

[2] A. Sarker, R. Zhang, Y. Wang, Y. Xiao, S. Das, D. Schutte, D. Oniani, Q. Xie, H. Xu, Natural language processing for digital health in the era of large language models, IMIA Yearb. Med. Inform. (2024). doi:10.1055/s-0044-1800750.

[3] A. Bornet, P. Khlebnikov, F. Meer, Q. Haas, A. Yazdani, B. Zhang, P. Amini, D. Teodoro, Analysis of eligibility criteria clusters based on large language models for clinical trial design, J. Am. Med. Inform. Assoc. 32 (3) (2025) 447–458. doi:10.1093/jamia/ocae311.

[4] J. C. L. Ong, B. J. J. Seng, J. Z. F. Law, L. L. Low, A. L. H. Kwa, K. M. Giacomini, D. S. W. Ting, Artificial intelligence, chatgpt, and other large language models for social determinants of health: current state and future directions, Cell Rep. Med. 5 (2024) 101356. doi:10.1016/j.xcrm.2023.101356.

[5] A. Neveditsin, et al., A comprehensive review of language models in medicine, PLOS Digit. Health 4 (5) (2025) e0000800. doi:10.1371/journal.pdig.0000800.

[6] M. Grunde-McLaughlin, M. S. Lam, R. Krishna, D. S. Weld, J. Heer, Designing LLM chains by adapting techniques from crowdsourcing workflows, ACM Trans. Comput.-Hum. Interact. 32 (3) (2025) 27. doi:10.1145/3716134.

[7] G. Li, H. A. A. K. Hammoud, H. Itani, D. Khizbullin, B. Ghanem, Camel: Communicative agents for "mind" exploration of large language model society, arXiv preprint arXiv:2303.17760 (2023).

[8] R. Polikar, Ensemble based systems in decision making, IEEE Circuits Syst. Mag. 6 (3) (2006) 21–45. doi:10.1109/MCAS.2006.1688199.

[9] V. Harish, et al., Teaching old tools new tricks—preparing emergency medicine for the impact of machine learning-based risk prediction models, CJEM 25 (5) (2023) 365–369. doi:10.1007/s43678-023-00480-8.

[10] Y. Chen, B. Wen, F. Zulkernine, A multiagent summarization and auto-evaluation framework for medical text: Development and evaluation study, JMIR AI 4 (2025) e75932. doi:10.2196/75932.

[11] R. Perera, A. Basnayake, M. Wickramasinghe, Autoscaling LLM-based multi-agent systems through dynamic integration of agents, Front. Artif. Intell. (2025). doi:10.3389/frai.2025.1638227.

[12] D. Dua, C. Graff, Heart disease data set [dataset], UCI Machine Learning Repository, accessed 2026 (1988). URL https://archive.ics.uci.edu/dataset/45/heart+disease

[13] UCI Machine Learning Repository, Pima indians diabetes database [dataset], Kaggle, originally from the UCI Machine Learning Repository; Accessed 2026 (2016).
URL https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

[14] L. Wynants, B. Van Calster, G. S. Collins, R. D. Riley, G. Heinze, E. Schuit, M. M. J. Bonten, J. A. A. Damen, T. P. A. Debray, M. De Vos, et al., Prediction models for diagnosis and prognosis of covid-19: systematic review and critical appraisal, BMJ 369 (2020) m1328. doi:10.1136/bmj.m1328.

[15] W. Fedus, B. Zoph, N. Shazeer, Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, J. Mach. Learn. Res. 23 (120) (2022) 1–39, preprint: https://doi.org/10.48550/arXiv.2101.03961.
URL https://jmlr.org/papers/v23/21-0998.html

[16] P. Zunjare, M. S. Hsiao, A hybrid multi-agent prompting approach for simplifying complex sentences, in: 2025 IEEE Smart World Congress (SWC), IEEE, 2025. doi:10.1109/SWC65939.2025.00070.

[17] Q. Kang, F. Wang, Z. Liu, Z. Chen, Special agents policy gradient in value decomposition-based approach, in: 2023 IEEE 12th Data Driven Control and Learning Systems Conf. (DDCLS), IEEE, 2023. doi:10.1109/DDCLS58216.2023.10165847.