ICDAGENT: Empowering Agentic Large Language Models for Explainable Medical Coding

Ziyi Yin[1], Yuanpu Cao[1], Ting Wang[2], Jinghui Chen[1], Fenglong Ma[1]*
[1]The Pennsylvania State University, [2]Stony Brook University
[1]{ziyiyin, ymc5533, jzc5917, fenglong}@psu.edu, [2]twang@cs.stonybrook.edu

Abstract

The explainable medical coding task aims to automatically assign International Classification of Diseases (ICD) codes to clinical notes while providing explicit justifications for each assignment. Recent approaches employ large language models (LLMs) to generate such explanations. However, their performance remains limited due to a lack of understanding of the clinical meanings of ICD codes. Additionally, the vast ICD code space further complicates the task of accurate prediction. To address these challenges, we propose the ICDAGENT framework, which consists of two collaborative LLM agents: a coding agent and a critical agent. The coding agent extracts ICD codes and generates preliminary rationales, while the critical agent performs fine-grained chain-of-thought reasoning to verify and refine them. Furthermore, the critical agent is trained with a rationale-aware reward, combined with reinforcement learning, enabling it to distinguish between correct and incorrect reasoning and ensure explanation accuracy. Experiments across multiple ICD coding standards and datasets demonstrate that ICDAGENT achieves effective ICD coding with accurate and trustworthy explanations. [1]

1 Introduction

The International Classification of Diseases (ICD) coding task aims to assign standardized codes from clinical notes, which is typically framed as a multi-label text classification problem, with numerous studies proposed to improve its predictive accuracy (Luo et al., 2024; Vu et al., 2020; Yuan et al., 2022). Despite the success of existing models, their lack of convincing, human-understandable explanations makes them difficult for physicians to trust and use in practice. To address this, the task of **explainable ICD coding** has emerged, aiming to provide justifications for each predicted code through methods like keyword-matching (Cheng et al., 2023; Douglas et al., 2025) and open-ended text generation (Li et al., 2024; Falis et al., 2024).

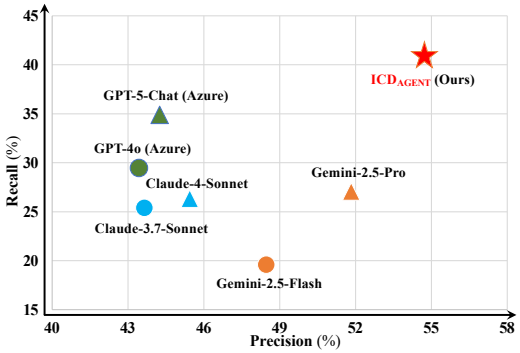

Figure 1: Comparison between existing proprietary LLMs and ICDAGENT in terms of precision and recall performance obtained using 1,000 samples from the MIMIC-IV dataset, where we use string match to identify the predicted ICD codes.

The goal of keyword matching is to identify words or text spans from clinical notes as explanatory evidence (Cheng et al., 2023; Douglas et al., 2025). This paradigm, however, is fundamentally limited as it merely highlights isolated words without explaining how this fragmented evidence coherently supports a specific ICD code. To overcome these limitations, several recent studies have explored using either a single strong Large Language Model (LLM) (Falis et al., 2024) or employing collaborations among multiple LLMs (Motzfeldt et al., 2025) to generate detailed rationales for ICD codes. However, as shown in Figure 1, current LLM-based approaches remain suboptimal, as these proprietary, general-purpose models inherently struggle with the performance of the medical coding task.

On one hand, these commercial LLMs lack specialized, clinical coding knowledge, as the use of patients' clinical records as training data is usually strictly restricted by privacy concerns (Nguyen et al., 2023b). On the other hand, the challenge comes from the complexity of ICD coding. It re-

*Corresponding author.
[1]The source code will be available at https://github.com/ericyinyzy/ICDAGENT.git.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14131–14149
July 2-7, 2026

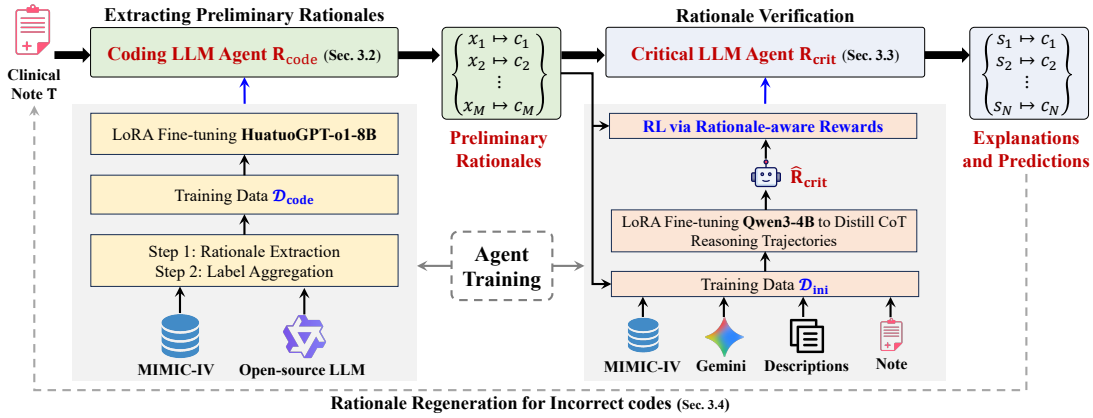

Figure 2: The overall structure of our proposed ICDAGENT framework.

quires LLMs to first identify critical medical evidence from unstructured clinical documents, and then map them to the correct codes. This challenge is further amplified by the vast ICD code space, where codes within the same category often have only subtle distinctions (Luo et al., 2024). Therefore, LLMs must perform fine-grained reasoning to ensure that the final explanations are accurate.

To overcome these challenges, we propose the ICDAGENT framework, which consists of a coding agent and a critical agent. As illustrated in Figure 2, the coding agent $\mathbf{R}_{\text{code}}$ takes a clinical note as input to predict ICD codes and generate a preliminary rationale $\mathbf{x}_i$ for each assigned code c_i. To inject specialized medical coding knowledge into $\mathbf{R}_{\text{code}}$, we construct a new coding rationale dataset and fine-tune the model through supervised fine-tuning. Recognizing that the preliminary predictions from $\mathbf{R}_{\text{code}}$ may contain errors, we introduce a second agent, the critical agent $\mathbf{R}_{\text{crit}}$, which performs Chain-of-Thought (CoT) reasoning to verify each rationale $\mathbf{x}_i$ and produce a final judgment on whether the corresponding code c_i is correct. To cultivate its reasoning capabilities, $\mathbf{R}_{\text{crit}}$ is first initialized by distilling CoT abilities from a proprietary teacher model and is subsequently optimized via reinforcement learning (RL). Furthermore, we propose a novel rationale-aware reward function that heuristically assigns learning incentives based on the quality of $\mathbf{x}_i$, thereby reducing verification errors. Ultimately, the ICDAGENT framework operates through a multi-round validation process between $\mathbf{R}_{\text{code}}$ and $\mathbf{R}_{\text{crit}}$ to produce the final ICD codes accompanied by their verified explanations.

We evaluated the performance of ICDAGENT on three datasets spanning both ICD-9 and ICD-10 versions. Our results demonstrate that ICDAGENT substantially outperforms existing commercial LLMs on all datasets. Additionally, ICDAGENT can be fully deployed on a local machine with two A100 GPUs, offering a more cost-efficient solution while simultaneously avoiding the privacy concerns associated with using commercial models. We hope that ICDAGENT can serve as an important direction for building more reliable and trustworthy AI in the clinical domain.

2 Related Work

Traditional Automatic ICD Coding.

Most current methods formulate the ICD coding task as a multi-label classification problem over a predefined, closed set of ICD codes (Mullenbach et al., 2018; Li and Yu, 2020; Cao et al., 2020; Vu et al., 2020; Yuan et al., 2022; Nguyen et al., 2023a; Luo et al., 2024; Huang et al., 2022; Michalopoulos et al., 2022; Liu et al., 2022). Although these multi-label classification methods can extract ICD codes with considerable accuracy, they cannot explain the reasoning or justification behind these extractions. As a result, this lack of explainability limits their use in real-world medical scenarios.

To address this limitation, a recent line of research has focused on explainable ICD coding (Mullenbach et al., 2018; Cheng et al., 2023; Douglas et al., 2025; Edin et al., 2024). Specifically, these methods augment the traditional classification task with a keyword-matching mechanism. Leveraging metrics such as attention scores (Douglas et al., 2025) or gradient-based influence (Edin et al., 2024), they identify salient keywords in the clinical note that are most relevant to a code's description, presenting these keywords as the rationale for their predictions. However, given the unique hierarchical structure of the ICD coding process, simply highlighting keywords cannot provide a systematic reasoning analysis for the target code, render-

ing the explanation insufficient in terms of comprehensiveness and accuracy. To overcome these limitations, we generate code explanations within an open-edned task in this paper, thereby offering more detailed and accurate explanations.

LLMs for Automatic ICD Coding. To provide more comprehensive justifications for extracted ICD codes, a promising approach is to leverage the advanced reasoning and generation capabilities of LLMs. One prominent strategy involves leveraging multiple proprietary LLMs to collaborate on extracting ICD codes (Li et al., 2024; Motzfeldt et al., 2025). For example, MAC (Li et al., 2024) employs multiple LLMs, powered by GPT-4o with distinct prompts, that simulate different roles such as an adjuster, physician, and patient. These agents interact with each other to reach a conclusion. Nevertheless, as previously discussed, proprietary LLMs inherently lack deep knowledge of ICD coding principles. Consequently, relying on such models for interaction does not effectively mitigate the risk of incorrect or omitted codes, leading to suboptimal performance. To address these challenges, we propose ICDAGENT, which coordinates specially fine-tuned LLMs to extract ICD codes, achieving superior coding performance while offering a more cost-effective solution.

3 Methodology

3.1 Overview

Given a clinical note $\mathbf{T}$ describing a patient's hospitalization, our task is to generate a set of pairs of explanation and ICD code, denoted by $[(\mathbf{s}_1, c_1), (\mathbf{s}_2, c_2), \cdots, (\mathbf{s}_N, c_N)]$. $\mathbf{s}_i$ is the explanation that identifies the relevant evidence extracted from $\mathbf{T}$ and demonstrates how this evidence supports the associated code c_i, thereby establishing a clear mapping $\mathbf{s}_i \mapsto c_i$[2]. N denotes the total number of predicted ICD codes.

To solve this task, we propose the ICDAGENT, a multi-agent LLM framework for ICD coding. As illustrated in Figure 2, ICDAGENT consists of two LLM agents: (1) a coding agent, $\mathbf{R}_{\text{code}}$, which is responsible for extracting ICD codes together with their preliminary rationales from $\mathbf{T}$; and (2) a critical agent, $\mathbf{R}_{\text{crit}}$, which evaluates the extracted rationales, discards invalid ones, and consolidates the valid rationales into coherent explanations $\mathbf{s}_i$ as the final explanation. Finally, $\mathbf{R}_{\text{code}}$ and $\mathbf{R}_{\text{crit}}$ are interacted over multiple rounds to extract all ICD codes and their explanations. In the following, we describe the training of both agents and explain how their collaboration yields the finalized ICD coding assignments.

[2]In ICDAGENT, each ICD code is represented using special delimiter tokens (e.g., |E11.9|). This format enables the corresponding codes to be easily extracted from output texts through string matching.

3.2 Training the Coding Model $\mathbf{R}_{\text{code}}$

Considering that ICD coding is a complex task requiring specialized clinical knowledge to map textual evidence in clinical notes to standardized codes, we propose a new LLM agent, $\mathbf{R}_{\text{code}}$, designed to extract all ICD codes along with their supporting rationales. We frame this new task as an open-ended text generation problem: given a clinical note $\mathbf{T}$, the model is trained to generate a comprehensive document $\mathbf{G}$, where it contains a series of rationales, each justifying a specific ICD code. Specifically, the textual structure of $\mathbf{G}$ can be formally formulated as follows:

$$\mathbf{G} = \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \cdots \oplus \mathbf{x}_M \oplus \mathbf{x}_c, \quad (1)$$

where each $\mathbf{x}_i$ is a preliminary rationale justifying a specific ICD code c_i, and $\oplus$ represents concatenation with a paragraph separator (e.g., "\n\n"). A specific sample is illustrated in Figure 8. $\mathbf{x}_c$ is a concluding sentence that provides a final summary of all extracted codes, thereby ensuring the overall structural integrity of the document. To accomplish this task, our methodology consists of two primary stages: first, the construction of a novel ICD-coding rationale dataset, $\mathcal{D}_{\text{code}}$; and second, the supervised fine-tuning of our agent $\mathbf{R}_{\text{code}}$ on this dataset. We describe each stage in detail below.

3.2.1 Data Collection of $\mathcal{D}_{\text{code}}$

The creation of $\mathcal{D}_{\text{code}}$ is developed on the MIMIC-IV dataset (Nguyen et al., 2023b) as it contains both ICD 9 and 10 codes. As a clinical dataset, it provides patients' clinical notes and their associated ICD codes, but does not explain why these codes are assigned. To address this problem, we propose a fully automated two-step labeling framework, detailed below.

Step 1: Rationale Extraction. Given a clinical note $\mathbf{T}$ and its ground truths with M ICD code labels $\mathcal{C}^\star = [c_1^\star, \ldots, c_M^\star]$, our first task is to create a golden rationale label $\mathbf{x}_i^\star$ for each correct code $c_i^\star$. Each rationale $\mathbf{x}_i^\star$ must identify the supporting evidence in $\mathbf{T}$ and justify why this evidence cor-

responds to its ground truth code. To accomplish this, we prompt a powerful open-source LLM.

The input to the LLM model comprises three components: the clinical note $\mathbf{T}$, the ground truth code $c_i^\star$, and the official textual description of $c_i^\star$[3]. By leveraging precise medical descriptions, the LLM can accurately locate key clinical cues, such as diagnoses, laboratory results, and surgical information, and integrate them into a coherent rationale. The final output of this step is a set of M textual rationales for the given note, represented as $[\mathbf{x}_1^\star, \ldots, \mathbf{x}_M^\star]$. The complete prompt can be found in Figure 5 and Appendix C. Experimental results in Appendix C and Table 6 further demonstrate the high reliability of the extracted rationales.

Step 2: Label Aggregation. Following the generation of individual rationales $[\mathbf{x}_1^\star, \ldots, \mathbf{x}_M^\star]$, we aggregate them into the final training label $\mathbf{G}^\star$ according to the structure defined in Eq. (1). To ensure the structural completeness of the generated output, we append a summary sentence $\mathbf{x}_c$ as the conclusion, which is constructed by arranging all identified codes into a sequence — for example, following the format: "*Therefore, the correct ICD codes are* $|c_1|$, ..., $|c_M|$." By repeating these two steps for each clinical note and its ICD codes, we construct the dataset of rationales for ICD coding, denoted as $\mathcal{D}_{\text{code}}$.

3.2.2 $\mathbf{R}_{\text{code}}$ Model Training with $\mathcal{D}_{\text{code}}$

After constructing $\mathcal{D}_{\text{code}}$, we perform supervised fine-tuning (SFT) on an open-source LLM. We adopt HuatuoGPT-o1-8B (Chen et al., 2024) as the backbone model, chosen for its extensive solid foundation of medical knowledge. During optimization, we adopt the low-rank adaptation training (LoRA) (Hu et al., 2021) to reduce the computational cost. Finally, we obtain $\mathbf{R}_{\text{code}}$, which is capable of generating ICD codes along with their preliminary rationales from the input note. The specific training prompts are provided in Figure 9 of Appendix H.

3.3 Training the Critical Model $\mathbf{R}_{\text{crit}}$

Given a clinical note, our coding model $\mathcal{R}_{\text{code}}$ generates a document $\mathbf{G} = \mathbf{x}_1 \oplus \mathbf{x}_2 \cdots \oplus \mathbf{x}_c$. After parsing $\mathbf{G}$, we can extract pairs of predicted codes and their corresponding rationales, e.g., c_i and $\mathbf{x}_i$. Ideally, this extracted rationale $\mathbf{x}_i$ could directly serve as the final explanation (i.e., $\mathbf{s}_i$), allowing us to set $\mathbf{s}_i = \mathbf{x}_i$. However, relying solely on $\mathbf{x}_i$ as the final explanation is insufficient, as these preliminary rationales may contain substantial hallucinations. For example, $\mathbf{x}_i$ may contain fabricated clinical evidence or rely on flawed reasoning to produce an incorrect code prediction.

To address this problem, we develop another LLM agent, $\mathbf{R}_{\text{crit}}$, to verify the correctness of each extracted code. Given a clinical note $\mathbf{T}$, a candidate rationale $\mathbf{x}_i$, and the associated code c_i, $\mathbf{R}_{\text{crit}}$ performs a step-by-step chain-of-thought (CoT) reasoning and results in a conclusion $\mathbf{y}_i$. If c_i is an incorrect code, $\mathbf{y}_i$ explains the error; otherwise, $\mathbf{y}_i$ refines $\mathbf{x}_i$ as the final explanation, i.e., $\mathbf{s}_i = \mathbf{y}_i$. A specific sample is illustrated in Figure 8.

To build $\mathbf{R}_{\text{crit}}$, we adopt a two-stage development process. Specifically, we first leverage a public LLM backbone and distill CoT reasoning trajectories from a proprietary teacher model via supervised fine-tuning (SFT), yielding an initial model $\hat{\mathbf{R}}_{\text{crit}}$. This step allows the model to acquire the necessary reasoning paths and output formats. Using $\hat{\mathbf{R}}_{\text{crit}}$ as initialization, we then further strengthen its critical reasoning ability through a heuristic reinforcement learning strategy, resulting in our final target model $\mathbf{R}_{\text{crit}}$. We describe each step in detail below.

3.3.1 Distilling an Initialized Agent $\hat{\mathbf{R}}_{\text{crit}}$

To create $\hat{\mathbf{R}}_{\text{crit}}$, we first construct a small dataset, $\mathcal{D}_{\text{ini}}$. Specifically, we sample a new subset of clinical notes from the MIMIC-IV dataset, which is non-overlapping with the data used for constructing $\mathcal{D}_{\text{code}}$. For each note $\mathbf{T}$, we adopt the well-trained $\mathbf{R}_{\text{code}}$ to extract all preliminary rationales $[\mathbf{x}_1 \cdots \mathbf{x}_N]$ and obtain their associated ICD codes $[c_1 \cdots c_N]$.

For each triple of $(\mathbf{T}, \mathbf{x}_i, c_i)$, together with the official description of c_i, we send them into the Gemini-2.5-Flash Thinking API and instruct it to perform a step-by-step reasoning to validate the correctness of c_i. The reasoning is formulated by two logical steps: first, verifying the factual accuracy of the evidence mentioned in $\mathbf{x}_i$, and second, assessing whether the reasoning based on this evidence appropriately leads to the target code. The API will output a reasoning trajectory and a conclusion text $\mathbf{y}_i^\star$. An example is shown in Figure 7 of Appendix I.

When c_i is incorrect, $\mathbf{y}_i^\star$ serves to explain the error; when correct, it further improves $\mathbf{x}_i$ to form the final explanation. To ensure quality, the reason-

[3]For example, the description of ICD 10 code "I50.9" is "*Heart failure, unspecified*".

ing trajectory and conclusion text are kept only if the decision of $\mathbf{y}_i^\star$ aligns with the ground truth correctness of c_i, determined by whether c_i belongs to the label set of ICD codes $\mathcal{C}^\star$. Finally, we combine the valid trajectory and conclusion into a single training label and add it to $\mathcal{D}_{\text{ini}}$.

Based on $\mathcal{D}_{\text{ini}}$, we adopt Qwen3-4B as the backbone LLM and apply SFT with LoRA training to obtain the initialized model $\hat{\mathbf{R}}_{\text{crit}}$. Given a large collection of ICD codes paired with their explanations, relying solely on SFT distillation is insufficient. To further enhance critical reasoning in unseen scenarios, we continue training $\hat{\mathbf{R}}_{\text{crit}}$ through reinforcement learning (RL).

3.3.2 RL via Rationale-aware Reward

Now the model $\hat{\mathbf{R}}_{\text{crit}}$ produces a binary judgment $\mathbf{y}_i$ (*Yes*/*No*) for each candidate code c_i. Drawing inspiration from existing verifiable RL work (DeepSeek-AI et al., 2025), a naïve reward is to directly compare this judgment with the ground-truth correctness of c_i, thereby promoting accurate verification. However, this simple reward has a significant loophole: it ignores the quality of the input rationale $\mathbf{x}_i$. In practice, $\mathbf{x}_i$ may include fabricated evidence or flawed reasoning yet still map to a correct code $c_i \in \mathcal{C}^\star$, thus leading to reward hacking, where the model learns to generate "*Yes*" without truly validating the evidence.

To solve this problem, we propose a rationale-aware reward r_{aware}. The core idea is to heuristically modulate the magnitude of rewards and penalties based on the quality of the input rationale x_i. Specifically, when a valid code c_i is derived from a high-quality rationale x_i, we amplify the feedback by increasing the reward for a correct judgment and the penalty for an incorrect one. Conversely, if a valid code c_i happens to be identified by chance but the supporting rationale x_i is significantly flawed, we reduce the magnitude of the corresponding feedback. This encourages the model to logically verify the code's correctness by detailed reasoning from x_i, rather than superficially matching the code itself. The function can be defined as follows:

$$r_{\text{aware}} = \begin{cases} 1 + Q_s, & \text{if } y_{\text{pred}} = 1, y_{\text{label}} = 1 \\ 1 + (1 - Q_s), & \text{if } y_{\text{pred}} = 0, y_{\text{label}} = 0 \\ -(1 - Q_s), & \text{if } y_{\text{pred}} = 1, y_{\text{label}} = 0 \\ -Q_s. & \text{if } y_{\text{pred}} = 0, y_{\text{label}} = 1 \end{cases}, \quad (2)$$

where Q_s is a quality score assigned to each input rationale $\mathbf{x}_i$, computed by Gemini-2.5-Flash before training. A higher score indicates better quality. The predicted correctness y_{pred} is obtained via a simple string-matching function, where 1 denotes that c_i is correct and 0 otherwise. The ground-truth correctness y_{label} is determined by whether c_i appears in the label set $\mathcal{C}^\star$. We employ Gemini-2.5-Flash to score each $\mathbf{x}_i$, and the specific scoring prompt is detailed in Appendix D.

Based on r_{aware}, we adopt Generalized Advantage Estimation (GAE) (Schulman et al., 2016) to estimate the advantage and update $\hat{\mathbf{R}}_{\text{crit}}$ with the proximal policy optimization (PPO) (Schulman et al., 2017) loss to ensure stable learning. Finally, we obtain the fine-tuned LLM $\mathbf{R}_{\text{crit}}$ as the critical agent, which takes $\mathbf{y}_i$ as a correct judgment (i.e., $y_{\text{pred}} = 1$) as well as the final explanation $\mathbf{s}_i$.

3.4 Multi-round Verification

Given a clinical note $\mathbf{T}$, we first use the coding model $\mathbf{R}_{\text{code}}$ to extract preliminary rationales $\{\mathbf{x}_i\}$ together with their candidate ICD codes $\{c_i\}$. We then apply $\mathbf{R}_{\text{crit}}$ to perform critical verification, yielding refined explanations $\{\mathbf{s}_i\}$ for accepted codes. This single pass filters out most incorrect codes, but it may also reject some valid codes when their accompanying rationales $\mathbf{x}_i$ are factually unsupported or logically flawed.

To promote both coverage and precision, we introduce a *multi-round* interaction between $\mathbf{R}_{\text{code}}$ and $\mathbf{R}_{\text{crit}}$ to finalize the predicted code set. Specifically, for the codes excluded in the previous round, we feed these codes together with their descriptions back to $\mathbf{R}_{\text{code}}$ to regenerate a rationale for each code, and then use $\mathbf{R}_{\text{crit}}$ to filter these rationales and add the codes that pass to the final set. In our experiments, we set the number of verification rounds to $K = 2$, striking a balance between performance gains and computational overhead. The sensitivity analysis of K is detailed in Appendix F. A more detailed analysis of the computational cost is presented in Appendix E.

4 Experimental Setups

Datasets. We conduct experiments under both the ICD-9 and ICD-10 coding standards and develop a corresponding version of our coding framework for each. For the ICD-9 scheme, we adopt the MIMIC-IV-ICD-9 training set, split from MIMIC-IV by following the setting of (Nguyen et al., 2023b). We use 15,000 clinical notes to train the coding model and 10,000 notes to train the critique model.

Among the 10,000 samples for training the critique model, 5,000 are used for initialization and the remaining 5,000 are used for reinforcement learning. For the ICD-10 scheme, we adopt the MIMIC-IV-ICD-10 training set (Nguyen et al., 2023b) with the same data size configuration. Additional implementation details are provided in Appendix A.

The evaluation is conducted on the test sets of MIMIC-III-ICD-9 (Mullenbach et al., 2018; Yuan et al., 2022), MIMIC-IV-ICD-9 (Nguyen et al., 2023b), and MIMIC-IV-ICD-10 (Nguyen et al., 2023b). Each test dataset is created by following its corresponding official setting. Given the substantial computational resources required for LLMs, we randomly sample 1,000 clinical notes from each test set and evaluate them using the corresponding version of our coding framework. During both training and testing, we consider the **full** set of ICD codes as prediction labels.

Baselines. This paper focuses on improving the performance of ICD coding with LLMs. To this end, we use two existing LLM-based approaches as baselines, including MAC (Li et al., 2024) and CLH (Motzfeldt et al., 2025), and their specific implementations are detailed in Appendix B. In addition, we include several of the latest proprietary LLMs as baselines, including Claude-4-Sonnet (Anthropic, 2025), Gemini-2.5-Pro (Comanici et al., 2025), and GPT-5-Chat (OpenAI, 2025). Note that all of these models are granted API query access by PhysioNet[4].

Evaluation Metrics. We conduct evaluations of different methods from two perspectives: *quantitative* metrics for ICD code prediction and *qualitative* metrics for rationale evaluation. For the *quantitative* evaluation, following prior work (Luo et al., 2024; Nguyen et al., 2023a), we report **Macro** and **Micro Precision**, **Recall**, and **F1** scores.

For the *qualitative* evaluation, we employ Gemini-2.5-Flash to assess the quality of each explanation. Specifically, for every predicted code, we provide its official description, the corresponding reasoning explanation, and the clinical note to the LLM, prompting it to select one of three options: *Incorrect*, *Partially Correct*, or *Fully Correct*.

Incorrect indicates that the predicted code itself is wrong or the explanation is entirely invalid (e.g., containing fabricated evidence). *Partially Correct* means the predicted code is correct and the explanation is mostly accurate, with only minor reasoning flaws. *Fully Correct* denotes a flawless explanation that clearly justifies the assignment of the correct ICD code. We report three metrics for explanation quality: the Fully Correct Ratio (**FCR**), i.e., the proportion of explanations rated *Fully Correct*; the Partially Correct Ratio (**PCR**), i.e., the proportion of explanations rated **at least** *Partially Correct*; and the Incorrect Ratio (**IR**), i.e., the proportion of explanations rated *Incorrect*.

Besides, we conduct human evaluation to manually assess the quality of the explanations by domain experts. The following two scores are used: (1) **Correctness Score**, which assesses whether the rationale provided by the model is factually accurate, and (2) **Completeness Score**, which measures whether the rationale sufficiently covers all relevant evidence in the patient's notes. Each score ranges from 1 to 5, with higher values indicating better performance.

Table 1: Quantitative evaluation results. All the compared methods are based on LLMs.

Method	Precision		Recall		F1	
	Macro	Micro	Macro	Micro	Macro	Micro
MIMIC-III-ICD-9						
MAC	21.02	46.50	19.84	33.78	20.41	39.13
CLH	21.66	42.53	16.59	18.10	18.79	25.39
Claude-4-Sonnet	24.52	56.04	18.11	27.9	20.83	37.28
Gemini-2.5-pro	21.52	57.04	17.36	26.81	19.22	36.48
GPT-5-Chat	22.09	56.57	16.09	25.56	18.61	35.21
ICDAGENT	**29.62**	**59.35**	**25.58**	**46.87**	**27.45**	**52.37**
MIMIC-IV-ICD-9						
MAC	21.94	43.70	28.15	38.37	22.52	40.86
CLH	23.67	41.23	19.50	19.85	21.38	26.80
Claude-4-Sonnet	27.92	53.24	23.42	33.81	25.48	41.36
Gemini-2.5-pro	24.01	54.39	20.24	30.89	21.97	39.40
GPT-5-Chat	23.64	54.85	19.80	30.96	21.56	39.58
ICDAGENT	**31.92**	**60.19**	**28.90**	**49.15**	**30.34**	**54.11**
MIMIC-IV-ICD-10						
MAC	21.82	44.77	21.89	35.06	21.86	39.32
CLH	21.70	43.42	18.63	24.74	20.05	31.52
Claude-4-Sonnet	18.02	45.44	15.16	26.33	16.46	33.34
Gemini-2.5-pro	21.99	51.82	17.97	27.09	19.78	35.58
GPT-5-Chat	21.58	44.52	21.36	35.57	21.47	39.55
ICDAGENT	**26.03**	**54.73**	**22.73**	**40.77**	**24.27**	**46.73**

5 Quantitative Results

5.1 Main Results

We first compare ICDAGENT across two ICD coding schemes and three datasets on quantitative metrics. The experimental results are shown in Table 1. From the table, we first observe that existing proprietary LLMs perform poorly, particularly on recall metrics. This finding supports our earlier claim that these models, lacking coding-specific knowl-

[4]Official Website: https://physionet.org/news/post/gpt-responsible-use. All commercial models other from Claude, GPT, and Gemini are prohibited from accessing the MIMIC dataset.

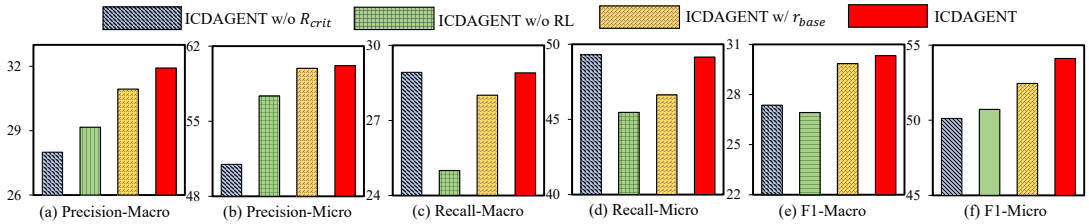

Figure 3: Ablation results on ICDAGENT. RL denotes reinforcement learning applied to $\mathbf{R}_{\text{crit}}$. r_{base} is the naive binary reward mentioned in Sec. 3.3.2, defined as $r_{\text{base}} = 1$ if $y_{\text{pred}} = y_{\text{label}}$, and 0 otherwise.

edge, tend to adopt an overly conservative coding strategy. In contrast, ICDAGENT substantially overcomes this limitation by achieving the best performance across all metrics. In addition, ICDAGENT demonstrates strong generalization capability: even without being trained on the MIMIC-III ICD-9 dataset, it still achieves the best prediction performance. These results highlight that ICDAGENT possesses stronger coding capability and can provide more accurate explanations for outputs.

5.2 Ablation Study Results

In ICDAGENT, we design two LLM agents: a coding agent $\mathbf{R}_{\text{code}}$ and a critical agent $\mathbf{R}_{\text{crit}}$. To fine-tune $\mathbf{R}_{\text{crit}}$, we introduce a rationale-aware reward function that heuristically assigns learning incentives. We evaluate the contribution of each component through the following ablations. "ICDAGENT w/o $\mathbf{R}_{\text{crit}}$" uses only $\mathbf{R}_{\text{code}}$ for prediction, i.e., $\mathbf{x}_i = \mathbf{s}_i$, where each $\mathbf{x}_i$ is parsed from the output of $\mathbf{R}_{\text{code}}$. "ICDAGENT w/o RL" combines $\mathbf{R}_{\text{code}}$ with a critical model trained solely via SFT on the distillation dataset $\mathbf{D}_{\text{ini}}$, without reinforcement learning. "ICDAGENT w/ r_{base}" trains the critical agent using a binary reward mentioned in Sec. 3.3.2, where $r_{\text{base}} = 1$ if the prediction y_{pred} of $\hat{\mathbf{R}}_{\text{crit}}$ matches the ground-truth correctness y_{label}, and 0 otherwise. We compare these variants with our proposed ICDAGENT.

All experiments are conducted on the MIMIC-IV-ICD-9 dataset, with results shown in Figure 3. We observe that introducing the critical agent notably improves coding accuracy. The training strategy of $\mathbf{R}_{\text{crit}}$ is also crucial: naive SFT or the binary reward r_{base} yields only limited precision gains but sharply reduces recall. We attribute this phenomenon to the limited accuracy of $\mathbf{R}_{\text{code}}$, which often generates codes and rationales with moderate errors. In such cases, the simple SFT or a binary reward makes the model overly strict and thus increases misjudgments. In contrast, ICDAGENT mitigates these issues with the rationale-aware reward and achieves superior performance.

6 Qualitative Evaluation

6.1 Main Results

Beyond improving the accuracy of code prediction, another key objective of our work is to provide clear and comprehensive explanations for ICD code assignments. To this end, we adopt Gemini-2.5-Pro as the evaluation tool to assess the quality of each explanation, and report the results in Table 2. As shown in the table, ICDAGENT achieves the best performance under both the stricter evaluation criterion (FCR) and the more flexible one (PCR & IR). In particular, under the ICD-10 standard, it demonstrates substantial quality improvements over most baselines. These results confirm that ICDAGENT is fully capable of generating accurate and comprehensive ICD code explanations.

Guideline-alignment evaluation. Beyond evaluating the quality of generated explanations solely from a clinical perspective, we further examine whether these explanations are aligned with official coding guidelines, which is more convincing for practical deployment. Specifically, the ICD-9-CM official coding book is organized into multiple chapters, each containing coding guidelines for a subset of ICD codes. For example, Chapter 1 covers the guidelines for codes |001|–|139|. To evaluate the alignment between model-generated explanations and the official coding guidelines, we first reorganized the guidelines in each chapter into a text passage. Then, for each clinical note, we provided the predicted ICD code explanations together with the corresponding coding guidelines to Gemini-2.5-Flash for explanation assessment. The evaluation criteria and prompt were largely consistent with those used in Section 6.1, except that we additionally required the LLM evaluator to judge whether each explanation was fully aligned, partially aligned, or not aligned with the coding guidelines. We report the results on the MIMIC-IV-ICD-9 dataset and compare our method against the three strongest baselines from Table 2. The results are presented in Table 3. From the table, we observe

Table 2: Qualitative evaluation results obtained through LLM-based assessment. FCR and PCR represent the proportions of fully correct and at least partially correct explanations, respectively (higher is better). IR represents the proportion of incorrect explanations or codes (lower is better).

Method	MIMIC-III-ICD-9			MIMIC-IV-ICD-9			MIMIC-IV-ICD-10		
	FCR↑	PCR↑	IR↓	FCR↑	PCR↑	IR↓	FCR↑	PCR↑	IR↓
MAC	47.43	47.98	52.02	44.02	45.50	54.50	43.94	44.49	55.51
CLH	43.89	44.01	55.99	42.01	43.37	56.63	42.36	43.06	56.94
Claude-4-Sonnet	54.93	55.62	44.38	52.26	52.96	47.04	44.80	45.35	54.65
Gemini-2.5-pro	56.24	56.88	43.12	54.08	54.27	45.73	51.41	51.71	48.29
GPT-5-Chat	55.63	56.42	43.58	54.24	54.94	45.06	43.60	44.05	55.95
ICDAGENT	**56.24**	**58.84**	**41.16**	**57.13**	**59.63**	**40.39**	**52.11**	**54.73**	**45.37**

Table 3: Evaluation of explanation alignment with official coding guidelines on the MIMIC-IV-ICD-9 dataset. Higher FCR and PCR indicate better performance, while lower IR is better.

Method	FCR (↑)	PCR (↑)	IR (↓)
Claude-4-Sonnet	49.42	51.43	48.57
Gemini-2.5-Pro	52.79	53.71	46.29
GPT-5-Chat	51.82	53.80	46.20
ICDAGENT	**56.41**	**58.93**	**41.07**

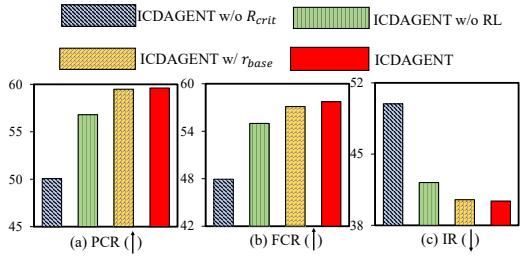

Figure 4: Ablation results of qualitative evaluation.

that ICDAGENT still substantially outperforms existing commercial LLMs. Moreover, comparing with Table 2, we find that although incorporating coding guidelines makes Gemini's evaluation more strict, ICDAGENT experiences the smallest drop in FCR scores. These findings demonstrate that our method's predicted explanations align more effectively with actual coding guidelines, validating ICDAGENT's broad application value.

Table 4: Human evaluation results. Each score ranges from 1 to 5. A higher score indicates better performance.

Method	Correctness	Completeness
MAC	4.65	3.20
CLH	4.35	2.65
Claude-4-Sonnet	4.85	3.55
Gemini-2.5-Pro	4.90	4.15
GPT-5-Chat	4.15	3.30
ICDAGENT	**4.95**	**4.20**

6.2 Human Evaluation Results

Both sets of results, as presented in Tables 1 and 2, clearly demonstrate the effectiveness of the proposed ICDAGENT. To further assess the quality of ICD code explanations, we conducted a human evaluation. Specifically, we first randomly selected ten ICD codes and identified clinical notes in the test set for which all six models produced correct predictions. For each selected code, we retained the corresponding clinical note, resulting in a total of 240 pairs of clinical notes and model-generated explanations. These pairs were then independently reviewed by medical experts, who rated each rationale based on two criteria: correctness and completeness. These codes are sampled equally from both the ICD-9 and ICD-10 standards.

We report the average scores for each dataset in Table 4. As shown in the table, ICDAGENT achieves the best overall performance, with particularly large gains in the completeness score compared to existing methods. This demonstrates the effectiveness of our coding LLM, which is able to more accurately identify relevant evidence in clinical notes and thereby generate explanations that are both comprehensive and specific.

6.3 Ablation Study Results

We report more ablation results on qualitative metrics evaluated by LLMs, including PCR, FCR and IR. The results are illustrated in Figure 4, and we observe the same pattern as in the quantitative results, where the integration of RL and the R_{crit} agent significantly enhances the quality of rationale prediction.

7 ICDAGENT *v.s.* Traditional Method

Although this work primarily focuses on generating explainable rationales for ICD coding, we also conducted experiments to compare ICDAGENT

Table 5: Results of CoRelation and ICDAGENT.

Method	Precision		Recall		F1	
	Macro	Micro	Macro	Micro	Macro	Micro
MIMIC-III-ICD-9						
CoRelation	23.28	**60.07**	22.52	**49.96**	22.90	**54.55**
ICDAGENT	**29.62**	59.35	**25.58**	46.87	**27.45**	52.37
MIMIC-IV-ICD-9						
CoRelation	24.85	57.26	27.47	**55.11**	26.10	**56.16**
ICDAGENT	**31.92**	**60.19**	**28.90**	49.15	**30.34**	54.11
MIMIC-IV-ICD-10						
CoRelation	14.49	54.41	15.92	**46.21**	15.17	**49.97**
ICDAGENT	**26.03**	**54.73**	**22.73**	40.77	**24.27**	46.73

with the state-of-the-art multi-label classification method, CoRelation (Luo et al., 2024), with the goal of evaluating predictive performance alone. To ensure fairness, CoRelation was trained on the same set of clinical notes used for ICDAGENT. As CoRelation is a closed-set multi-label classification method, we performed a quantitative comparison using all standard evaluation metrics, and the results are reported in Table 5.

From the table, we acknowledge that ICDAGENT is slightly weaker than CoRelation on micro-level metrics, which are dominated by common codes. We attribute this limitation to the fact that the two methods are targeted at different objectives. Specifically, common codes appear abundantly in the training data, allowing the classification model CoRelation to capture surface-level lexical patterns (e.g., keywords), resulting in a high recall. In contrast, ICDAGENT goes beyond code extraction and requires LLMs to construct explicit, evidence-grounded clinical rationales, which demands substantially more complex reasoning and verification. Even for the same common code, the supporting clinical evidence and rationales can vary widely across different notes, making the rationale patterns far less uniform than the classification signals exploited by traditional models. With limited training data, learning these diverse reasoning patterns naturally results in slightly lower recall on common codes than pure classifiers. This exact mechanism also explains why ICDAGENT outperforms CoRelation on macro-level metrics, which emphasize rare codes. When training $\mathbf{R}_{\text{code}}$, the inclusion of rationale generation forces the LLM to model the logical relationships between each rare code and the corresponding clinical evidence, thereby facilitating better generalization for long-tail cases. For traditional classifiers, the scarcity of rare codes makes it difficult to capture reliable lexical patterns automatically, leading to lower macro-level performance.

Discussion: The relative importance of Micro-F1 (common cases) vs. Macro-F1 (rare cases). Although both common and rare codes are important for automatic ICD coding, the existing evolution of ICD coding research shows that rare-code prediction is considerably more challenging and clinically valuable. Firstly, the ICD coding space is extremely large, and the vast majority of ICD codes are rare. For example, the top 10% most frequent codes in MIMIC-III account for 85% of all code occurrences (Zhou et al., 2021). As a result, experienced coders can typically assign these high-frequency codes reliably, whereas the rare codes, especially those with complex or nuanced descriptions, are substantially harder to identify. Moreover, the long-tail distribution of ICD codes is a well-recognized challenge and has been emphasized in many prior works (Zhou et al., 2021). Thus, ICDAGENT's stronger performance on rare codes, together with its ability to generate explicit and reliable rationales, provides meaningful and practical value: it can help practitioners effectively audit and verify difficult, low-frequency ICD codes. Taken together, we believe that although ICDAGENT is slightly lower than traditional classification models in Micro-F1, its high-quality rationale generation and strong overall coding performance still allow it to make a substantial contribution to this domain.

Extra experimental results. We also include additional experimental results in the appendix, which provide computational cost and hyperparameter sensitivity analysis, in Appendices E and F, respectively. We further discuss the effect of using different LLM backbones in Appendix G, and present a case study in Appendix I.

8 Conclusion

In this paper, we propose ICDAGENT, an LLM-based multi-agent framework for the ICD coding task. ICDAGENT first employs a coding LLM to extract ICD codes from clinical notes and generate their preliminary rationales. Next, a critical agent performs deep reasoning to filter out unreasonable predictions, guided by a novel rationale-aware reward combined with reinforcement learning to ensure reasoning accuracy. Finally, extensive experimental results demonstrate the effectiveness of the proposed ICDAGENT.

Acknowledgment. This work was partially supported by the National Science Foundation under Grant No. 223827 (F. Ma), 2405136 (T. Wang), and 2406572 (T. Wang).

Limitations. The limitations of this work can be summarized in two aspects. First, compared with traditional state-of-the-art multi-label classification methods, our approach still shows a gap in coding performance, particularly in terms of recall. Thus, further improving the ability of LLMs to extract ICD codes remains an important research direction. Second, the current training process of ICDAGENT requires a large amount of annotated data to inject ICD coding knowledge. While effective, such annotation is undoubtedly time-consuming and labor-intensive, and relying on commercial model APIs also introduces considerable financial burden. To address these limitations, one potential solution is to expand the official ICD code descriptions by incorporating detailed coding guidelines, rather than relying on their current short phrase definitions. These enriched descriptions could then be integrated into the model's prompts through approaches such as retrieval-augmented generation (RAG), thereby improving coding performance while avoiding costly training procedures. Given that building such a resource requires specialized design and data collection, we regard this as an important direction for future work.

Ethical Statement. In this paper, we propose ICDAGENT, an LLM-based multi-agent framework for ICD code prediction. We conduct extensive experiments on the MIMIC-III and MIMIC-IV datasets. To comply with the data usage agreements of the MIMIC datasets and to protect patient privacy, we confirm that all commercial LLMs used in this study were accessed through APIs with authorized access. We envision ICDAGENT as a promising direction for advancing automatic ICD coding, particularly in enabling users to obtain more comprehensive and logically sound coding analyses.

AI assistants in this research. We only adopt the AI assistant tool at the sentence level for fixing grammar and polishing sentences.

References

Anthropic. 2025. Introducing claude 4. https://www.anthropic.com/news/claude-4. Accessed: April 18, 2026.

Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. 2020. Hypercore: Hyperbolic and co-graph representation for automatic ICD coding. In *ACL*, pages 3105–3114. Association for Computational Linguistics.

Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. 2024. Huatuogpt-o1, towards medical complex reasoning with llms. *CoRR*, abs/2412.18925.

Hua Cheng, Rana Jafari, April Russell, Russell Klopfer, Edmond Lu, Benjamin Striner, and Matthew Gormley. 2023. MDACE: MIMIC documents annotated with code evidence. In *ACL (1)*, pages 7534–7550. Association for Computational Linguistics.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, and S. S. Li. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

James C. Douglas, Yidong Gan, Ben Hachey, and Jonathan K. Kummerfeld. 2025. Less is more: Explainable and efficient ICD code prediction with clinical entities. In *ACL (1)*, pages 30835–30847. Association for Computational Linguistics.

Joakim Edin, Maria Maistro, Lars Maaløe, Lasse Borgholt, Jakob D. Havtorn, and Tuukka Ruotsalo. 2024. An unsupervised approach to achieve supervised-level explainability in healthcare records. In *EMNLP*, pages 4869–4890. Association for Computational Linguistics.

Matús Falis, Aryo Pradipta Gema, Hang Dong, Luke Daines, Siddharth Basetti, Michael Holder, Rose S. Penfold, Alexandra Birch, and Beatrice Alex. 2024. Can GPT-3.5 generate and code discharge summaries? *J. Am. Medical Informatics Assoc.*, 31(10):2284–2293.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Chao-Wei Huang, Shang-Chi Tsai, and Yun-Nung Chen. 2022. PLM-ICD: automatic ICD coding with pretrained language models. *CoRR*, abs/2207.05289.

Fei Li and Hong Yu. 2020. ICD coding from clinical text using multi-filter residual convolutional neural network. In *AAAI*, pages 8180–8187. AAAI Press.

Rumeng Li, Xun Wang, and Hong Yu. 2024. Exploring LLM multi-agents for ICD coding. *CoRR*, abs/2406.15363.

Leibo Liu, Óscar Pérez Concha, Anthony N. Nguyen, Vicki Bennett, and Louisa Jorm. 2022. Hierarchical label-wise attention transformer model for explainable ICD coding. *J. Biomed. Informatics*, 133:104161.

Junyu Luo, Xiaochen Wang, Jiaqi Wang, Aofei Chang, Yaqing Wang, and Fenglong Ma. 2024. Corelation: Boosting automatic ICD coding through contextualized code relation learning. In *LREC/COLING*, pages 3997–4007. ELRA and ICCL.

George Michalopoulos, Michal Malyska, Nicola Sahar, Alexander Wong, and Helen H. Chen. 2022. Icdbigbird: A contextual embedding model for ICD code classification. In *BioNLP@ACL*, pages 330–336. Association for Computational Linguistics.

Andreas Motzfeldt, Joakim Edin, Casper L Christensen, Christian Hardmeier, Lars Maaløe, and Anna Rogers. 2025. Code like humans: A multi-agent solution for medical coding. *arXiv preprint arXiv:2509.05378*.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. In *NAACL-HLT*, pages 1101–1111. Association for Computational Linguistics.

Thanh-Tung Nguyen, Viktor Schlegel, Abhinav Ramesh Kashyap, and Stefan Winkler. 2023a. A two-stage decoder for efficient ICD coding. In *ACL (Findings)*, pages 4658–4665. Association for Computational Linguistics.

Thanh-Tung Nguyen, Viktor Schlegel, Abhinav Ramesh Kashyap, Stefan Winkler, Shao-Syuan Huang, Jie-Jyun Liu, and Chih-Jen Lin. 2023b. Mimic-iv-icd: A new benchmark for extreme multilabel classification. *CoRR*, abs/2304.13998.

OpenAI. 2025. Gpt-5 system card. https://cdn.openai.com/gpt-5-system-card.pdf. Technical report describing GPT-5 safety evaluations and capabilities. Accessed: April 18, 2026.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. 2016. High-dimensional continuous control using generalized advantage estimation. In *ICLR (Poster)*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297.

Thanh Vu, Dat Quoc Nguyen, and Anthony N. Nguyen. 2020. A label attention model for ICD coding from clinical text. In *IJCAI*, pages 3335–3341. ijcai.org.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jian Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report. *CoRR*, abs/2505.09388.

Zheng Yuan, Chuanqi Tan, and Songfang Huang. 2022. Code synonyms do matter: Multiple synonyms matching network for automatic ICD coding. In *ACL (2)*, pages 808–814. Association for Computational Linguistics.

Tong Zhou, Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Kun Niu, Weifeng Chong, and Shengping Liu. 2021. Automatic ICD coding via interactive shared representation networks with self-distillation mechanism. In *ACL/IJCNLP (1)*, pages 5948–5957. Association for Computational Linguistics.

A Implementation Details

All experiments are conducted on two NVIDIA A100 GPUs, each equipped with 80GB of memory. For training the coding model, we use **HuatuoGPT-O1-8B** (Chen et al., 2024) as the backbone, as it has been pre-trained on large-scale professional medical corpora. We perform LoRA-based fine-tuning with a LoRA rank of 32 for 3 epochs, using a typical learning rate of 3×10^{-4}. For training the critique model, we adopt **Qwen3-4B** (Yang et al., 2025) as the backbone. The initialization stage is conducted with a LoRA rank of 32 and trained for 3 epochs on $\mathcal{D}_{\text{ini}}$, using a learning rate of 3×10^{-4}. Finally, we employ the VERL framework (Sheng et al., 2025) for reinforcement learning and perform full-parameter fine-tuning with a learning rate of 1×10^{-6} for 2 epochs. The entire training process can be completed within 72 GPU hours.

B Details of Baseline Methods

Multi-Agent Coding (MAC) framework (Li et al., 2024). The MAC method employs five proprietary LLMs to mimic the human process of extracting ICD codes. Specifically, a *coder agent* is first adopted to identify ICD codes from clinical notes as thoroughly as possible. The extracted codes, along with their explanations, are then passed to a *reviewer agent* for filtering. The filtered results are subsequently sent to two additional agents, which act as a *patient* and a *physician*, respectively, to further review the coding decisions. If both agents agree on the current coding, the codes are output as the final predictions; otherwise, the system invokes an additional agent to make the final decision. We follow their original settings and prompts in our experiments.

Code Like Humans (CLH) framework (Motzfeldt et al., 2025). The CLA framework adopts a RAG-like architecture consisting of four agents. First, an LLM-based *evidence extractor* is employed to identify codeable conditions by extracting text snippets from the original clinical notes. Given each text snippet as input, a *retrieval agent* matches it to valid coding terms in the alphabetical index. The retrieved results are then passed to an *LLM-based verification agent*, which narrows down the candidate codes by applying formal coding rules. Finally, another *LLM agent* reviews these codes to finalize the coding assignments. We follow their official implementation, with one modification: in the final review agent, we add a requirement that the model must provide explicit reasoning for each code assignment, enabling a fair comparison with our method.

C Details of $\mathcal{D}_{\text{code}}$ Creation

The coding rationale dataset $\mathcal{D}_{\text{code}}$ is created for fine-tuning our coding LLM agent $\mathbf{R}_{\text{code}}$. To create $\mathcal{D}_{\text{code}}$, we first collect 15,000 clinical notes and their ICD codes from the MIMIC-IV dataset. We apply the public Qwen2.5-72B model to extract the ground-truth rationale $\mathbf{x}_i^\star$ for each sample. The specific input prompt is detailed in Figure 5, where we explicitly request Qwen2.5-72B to generate only one paragraph for each ICD code, and the ICD code must be formatted in "|ICD_code|" for the string matching purpose.

Table 6: Agreement ratio between Qwen2.5-72B and Gemini-2.5-Flash. We adopt Qwen2.5-72B to generate rationales, and use Gemini-2.5-Flash to evaluate them. IR(↓) denotes the ratio of incorrect rationales.

	FCR(↑)	PCR(↑)	IR(↓)
Quality of $\mathbf{x}_i^\star$	90.97	92.14	7.86

Quality of the Generated Rationale $\mathbf{x}_i^\star$. To ensure that the generated rationales are effective for training our coding LLM $\mathbf{R}_{\text{code}}$, we conduct an additional experiment to verify their quality. Specifically, we employ Gemini-2.5-Flash to evaluate each ground-truth rationale $\mathbf{x}_i^\star$ given the corresponding clinical note. The LLM is required to assess the correctness of the input rationales and classify them into one of three categories: Fully Correct, Partially Correct, or Incorrect. This evaluation is performed on 1,500 ICD codes and their associated rationales. We report the ratio of each choice, as summarized in Table 6. From the results, we observe that our generated rationales $\mathbf{x}_i^\star$ are highly reliable, with a fully correct rate of 90.97%. When including partially correct cases, the overall validity reaches 92.14%. These findings demonstrate the effectiveness of our constructed coding rationale dataset.

D Scoring Rubric for Generating Q_s

To train the critical agent $\mathbf{R}_{\text{crit}}$, we propose a rationale-aware reward r_{aware}. Specifically, r_{aware} dynamically assigns learning incentives according to the quality score Q_s of the input rationale $\mathbf{x}_i$. Here, $Q_s \in \{0.2, 0.4, 0.6, 0.8\}$ is obtained from

model: ***Qwen2.5-72B***
messages = [{
"role": System,
"content": Now you are a professional ICD code annotator. Given a patient's clinical note from their hospital stay. Your task is to write a text to explain the reasoning behind its association with a target ICD code.
}
{"role": User,
"content": Here are my inputs:
INPUT 1: The patient's clinical note: {clinical_note}. "
INPUT 2: Your target ICD code {|ICD_code|} and its official description: {target_code_description}
Now your task is to write a rational in a way that logically leads to the final target ICD code. Please follow these requirements:
1. Generate only **one paragraph**.
2, Your rational need strictly base on the evidence provided in the clinical note — do not add imaginative or fabricated content.
3.Keep your written accurate, clear, and concise.
4. All ICD codes must appear in the format |ICD_code|. Never output a bare code with out | |
5. Now, please generate your rational:
}]

Figure 5: The prompt for generating rationale.

model: ***Gemini-2.5-Flash***
messages = [{
"role": System,
"content": You are a meticulous AI expert in medical ICD coding auditing. Your objective is to critically assess the quality of a given rationale that justifies the extraction of an ICD code from a clinical note.
}
{"role": User,
"content": ### INPUTS:
INPUT 1: the clinical note {clinical_note}.
INPUT 2: Rationale for Evaluation: {Initial Rationale from the Code LLM Agent}.
INPUT 3: Your target code and its official description: {|ICD_code|}:{target_code_description}
Based on the inputs above, you must evaluate the rational for the extracted code and assign a score based on the rubric below. You need score the rational provided in INPUT 2.
SCORING RUBRIC:
0.2 : The rationale cite evidence that does not exist in the clinical note, and the logical connection to the code is completely flawed, irrelevant, or medically incorrect.
0.4: The rationale cites valid evidence from the note, and the reasoning is on the right track but contains significant flaws. It might overlook crucial context, misinterpret information, or have major logical gaps.
0.6: The rationale is well-founded, citing correct evidence and providing a logical argument. However, it may contain minor inaccuracies, non-critical omissions, or could be slightly more precise.
0.8: The rationale is perfect. It cites all necessary and correct evidence from the clinical note, and the reasoning is logically sound, complete, and medically accurate, leaving no room for ambiguity
Now, begin your evaluation. First output your score, and then output a concise reason }]

Figure 6: The prompt and specific scoring rubric for generating Q_s.

the output of an LLM, where a higher score indicates better rationale quality. The detailed scoring rubric for each level, along with the specific LLM prompt, is shown in Figure 6. To ensure accurate scoring, we provide the LLM with the corresponding clinical note, the associated ICD code, and its official description as references for evaluation. Finally, we employ the Gemini-2.5 API to assign Q_s to each input rationale.

E Computational Cost Evaluation

In real-world ICD coding scenarios, users often need to process and validate a large number of clinical notes at the same time. Therefore, the efficiency of generating ICD codes should also be considered as an important aspect to highlight the practicality of different methods. Specifically, we measure runtime efficiency by computing the average prediction time per 100 codes for all baseline methods. All experiments are conducted on the MIMIC-IV-ICD-9 dataset. For a more intuitive comparison, we also include the F1-Micro and F1-Macro scores from Table 1 as reference. Experimental results are illustrated in Table 7.

GPT-5-Chat achieves the fastest computational efficiency, as it does not require deep reasoning when generating answers. ICDAGENT attains the second-highest efficiency while maintaining the best predictive performance. We attribute this to its use of smaller, lightweight LLMs for both coding

and verification. In addition, our method can be readily deployed under limited computational resources, requiring only two A100 GPUs to ensure stable operation, thereby providing a more practical solution for automatic ICD coding.

Table 7: Computational efficiency comparison. We report the average runtime for processing every 100 clinical notes.

Method	F1		Time
	Micro	Macro	(min/100)
MAC	22.52	40.86	98.45
CLH	21.38	26.80	66.51
Claude-4-Sonnet	25.48	41.36	34.29
Gemini-2.5-pro	21.97	31.40	86.35
GPT-5-Chat	21.56	39.58	**17.62**
ICDAGENT	**30.34**	**54.11**	32.90

F Hyperparameter Sensitivity Analysis

In the ICDAGENT framework, the coding LLM ($\mathbf{R}_{\text{code}}$) and the critical LLM ($\mathbf{R}_{\text{crit}}$) interact across multiple rounds to generate the final ICD code predictions. The number of interaction rounds, denoted as K, is treated as a hyperparameter, and in this section we investigate how different values of K affect prediction performance. Specifically, we set $K = 1, 2, 3$ and compare their results across all evaluation metrics, as shown in Table 8. All experiments are conducted on MIMIC-IV-ICD-9. From the table, an important observation emerges: when no multi-round verification is used (i.e., $K = 1$), the model achieves higher precision but suffers from lower recall, since some rationales for the codes are incorrect. By contrast, with two rounds of verification, the recall and F1 scores improve substantially while the average explanation score remains nearly unchanged. Furthermore, increasing the number of rounds to three ($K = 3$) does not yield significant additional gains, so we set $K = 2$ in our experiments.

Multi-round recovery analysis. To better investigate how missed valid codes in the first round affect the final results, we next conducted a recovery analysis to examine whether these codes can be captured in subsequent rounds. Specifically, for the test split of MIMIC-IV-ICD-9, there are 7,455 valid codes that were missed after the first round. The recovery code numbers in the subsequent rounds are reported in Table 10.

From the experimental results, we observe that 11.7% of the valid codes missed in the first round can be recovered in the second round, thereby improving the overall F1 score. However, when the number of rounds increases (e.g., K=3), this incremental gain becomes negligible. This phenomenon is consistent with our previous observations.

model's response:
<think> **Analyzing the ICD-10 Code E11.9** Okay, here's my assessment of this coding scenario. .....
First, I need to confirm the evidence. The note needs to explicitly mention "diabetes mellitus type 2 (DMII)" and that the patient was on an insulin sliding scale.
I'm looking for direct quotes or very close paraphrases.
Bingo! I see "NIDDM" in the History of Present Illness , which is a common abbreviation for Type 2 Diabetes Mellitus. **.....**
Next, I need to evaluate the reasoning. Does the evidence logically connect to the code E11.9?
The note clearly mentions "DMII" and "Diabetes (diet controlled)," which supports Type 2 Diabetes. **.....**
So, in conclusion, all the evidence is there. The reasoning makes sense. I think E11.9 is the correct code based on the information provided, given the constraints of the prompt. It's a clean fit. **</think>**

|E11.9| is **a correct ICD code** because the clinical note explicitly mentions "DMII: Diet controlled. On insulin sliding scale while in hospital" under the "CHRONIC ISSUES" section, **.....**

Figure 7: An example of a thinking trajectory to obtain the correctness judgement.

G Performance based on Different LLM Backbones

In our experiments, we adopt HuatuoGPT-o1-8B as the backbone to fine-tune the coding LLM agent $\mathbf{R}_{\text{code}}$. To further evaluate the generalization ability of ICDAGENT under different backbones, we also compare its performance when using Qwen3-8B and Qwen3-4B as alternatives. All experiments are conducted on the MIMIC-IV-ICD-9 dataset. The results are reported in Table 9. From the results, we observe two key findings: (1) HuatuoGPT-o1-8B achieves the best performance, which can be attributed to its extensive post-training on medical-domain corpora, enabling better adaptation to clinical coding tasks. (2) Our training approach exhibits strong generalization capability, as even the smaller Qwen3-4B model attains competitive coding performance after fine-tuning. We further extend this experiment by training different critical agents. Specifically, we conduct experiments using the smaller Qwen3-1.7B model as the backbone. The results show that our method still achieves strong predictive performance, further demonstrating the robust generalization ability of ICDAGENT, which enables its deployment even on resource-constrained platforms.

Round Number K	Quantitative Metrics						Qualitative Metrics		
	Precision		Recall		F1		FCR	PCR	IR
	Macro	Micro	Macro	Micro	Macro	Micro			
1	30.04	62.15	23.62	42.43	26.44	50.43	58.83	61.40	38.60
2	31.92	60.19	28.90	49.15	30.33	54.11	57.13	59.61	40.39
3	31.88	59.03	28.98	49.48	30.36	54.09	56.95	58.52	41.48

Table 8: Hyperparameter Sensitivity Analysis

Agent Type	Backbone Model	Coding Performance						Explanation Quality		
		Precision		Recall		F1		FCR	PCR	IR
		Macro	Micro	Macro	Micro	Macro	Micro			
$\mathbf{R}_{\text{code}}$	Qwen3-4B	27.70	50.31	26.50	47.17	27.09	48.69	46.29	49.24	50.76
	Qwen3-8B	28.66	50.47	27.96	48.49	28.31	49.46	47.48	49.92	50.08
	HuatuoGPT-o1	28.00	50.98	27.72	49.30	27.36	50.12	47.94	50.06	49.94
$\mathbf{R}_{\text{crit}}$	Qwen3-1.7B	30.31	55.01	28.80	49.09	29.58	51.88	49.74	52.95	47.05
	Qwen3-4B	31.92	60.19	28.90	49.15	30.33	54.11	57.13	59.61	40.39

Table 9: Ablation experiments with different LLM backbones.

Table 10: Number of recovered codes under different verification rounds K.

K	**Recovered Codes**
2	870
3	82

Preliminary Rationale x_i	Final Explanation y_i
The patient has a history of diabetes mellitus type 2 (DMII), which is diet-con trolled, as noted in the Past Medical History section. During the hospital stay, the patient was on an insulin sliding scale, indicating active management of blood glucose levels. These details support the assignment of the ICD code |E11.9| for Type 2 diabetes mellitus without complications.	|E11.9| is a correct ICD code because the clinical note explicitly mentions "DMII: Diet controlled. On insulin sliding scale while in hospital" under the "CHRONIC ISSUES" section, and lists "Insulin SC Sliding Scale" under "Medications on Admission." It directly supports Type 2 Diabetes Mellitus, and the mention of an insulin sliding scale during the hospital stay indicates active management of blood glucose levels, which is consistent with the code |E11.9| for Type 2 diabetes mellitus without complications.
The patient's clinical note mentions a history of coronary artery disease (CAD) and a recent coronary artery bypass graft (CABG) procedure involving four grafts. This indicates the presence of significant atherosclerotic heart disease affecting the native coronary arteries, which aligns with the ICD code |I25.10| for atherosclerotic heart disease of native coronary artery without angina pectoris. The absence of reported angina pectoris during the hospital stay further supports the use of this specific code.	|I25.10| is a correct ICD code because the clinical note explicitly mentions "history of coronary artery disease (CAD)" and " recent CABG x4" under both the "History of Present Illness" and "Brief Hospital Course" sections, and lists "CAD s/p CABG x4" as a secondary diagnosis. This evidence directly supports the presence of atherosclerotic heart disease of native coronary arteries. Furthermore, the note explicitly states in the "Brief Hospital Course" that "Throughout the hospitalization, the patient denied chest pain, shortness of breath, anginal symptoms," which directly supports the "without angina pectoris" component of the code.

Figure 8: Examples of preliminary rationales and final explanations.

H Other LLM Prompts

For generating the preliminary rationales with $\mathbf{R}_{\text{code}}$, we adopt the LLM prompt shown in Figure 9, which explicitly instructs the model to output the ICD codes together with their corresponding rationales in the required format. For the prompts used as input to the critical agent $\mathbf{R}_{\text{crit}}$, we provide them in Figure 10. These prompts explicitly guide the model to follow a two-step verification process for ICD coding: first, to check whether the cited evidence truly exists in the clinical note, and second, to assess the plausibility of the explanation.

I Case Study

We first present the reasoning trajectories and conclusions obtained using the Gemini-2.5-flash think API. This process is illustrated in Figure 7. As shown in the figure, the Gemini-2.5-flash model performs fine-grained, step-by-step reasoning to reach the final correctness judgment. The reasoning consists of two main steps: (1) verifying whether the cited evidence actually exists in the clinical note, and (2) assessing whether this evidence can accurately support the target ICD code. Through these two reasoning stages, the model determines whether the predicted code c_i is correct. The output displayed in Figure 7 corresponds to the final output of $\mathbf{R}_{\text{crit}}$, where $\mathbf{y}_i$ denotes the final explanation. We also present several examples of preliminary rationales $\mathbf{x}_i$ and their corresponding final explanations $\mathbf{y}_i$, as illustrated in Figure 8.

Finally, we visualize several ICD code prediction results in Figure 11 and Figure 12. Each figure presents the explanations extracted for a single code under both ICD-9 and ICD-10. Since clinical

model: ***Code LLM Agent***
messages = [{
"role": System,
"content": You are a helpful assistant specialized in ICD coding. You will be given a clinical note, and your task is to extract the correct ICD codes that match the clinical note. For each extracted code (formatted as '|ICD_code|'), you must:
- Identify the specific clues in the clinical note that support the use of this code.
- Explain in detail how these clues justify the target code. }
{"role": User,
"content": Here is the clinical note: {clinical_note}. Predict the correct ICD codes based on the given note. You must follow these output formats:
1. Each section is a clue and a reasoning to the correct codes.
2. Seperate each section with '\n\n'.
3. For the output format, you must directly and strictly output N paragraphs, which contain the rationals of N codes.
4. Don't forget to include a conclusion sentence, for example 'Therefore, the correct icd codes are |ICD_code1||ICD_code2|...(Your extracted codes.)'
}]

Figure 9: The prompt of $\mathbf{R}_{\text{code}}$.

model: ***Gemini-2.5-Flash***
messages = [{
"role": System,
"content": You are an expert medical ICD-10 coding auditor. Your mission is to rigorously audit a predicted ICD code based on a patient's clinical note and the reasoning clues provided by a model. You must perform a structured, multi-step validation:
1. Evidence Verification: Confirm that every reasoning clue is explicitly present in the clinical note. The evidence must be a direct quote or a very close paraphrase.
2. Reasoning & Justification: Ensure the link between the evidence and the inferred condition is medically sound and meets the official definition for the ICD code.}
{"role": User,
"content": Here is the clinical note: {clinical_note}.
The target candidate ICD Code you need to judge: {|ICD_code|}:{target_code_description}
The model's predicted rationale on this target code: {Initial Rationale from the Code LLM Agent}
Please think step by step to formulate your final judgment. Your final output must strictly these formats:
1. If the code is correct: |ICD_Code| is a correct ICD code because the clinical note mentions...(refine the rationale and give your refined explanation)
2. If the code is incorrect: |ICD_Code| is not a correct ICD code because... (state the specific reason)
Now perform your reasoning:
}]

Figure 10: The prompt for generating reasoning analysis on the preliminary rationale. We adopt the same prompt for $\mathbf{R}_{\text{crit}}$ and Gemini-2.5-Flash API.

notes are typically very long, we only include selected key contexts and highlight relevant evidence. From these examples, we observe that compared with existing proprietary LLMs, ICDAGENT is able to identify more comprehensive clinical evidence. For instance, in Figure 11, ICDAGENT not only captures the diagnostic information in the Brief Hospital Course section, but also accurately locates numerical results from the Transthoracic Echocardiogram (TTE) as supporting evidence. As a result, it can provide more precise explanations, which further demonstrates the effectiveness of ICDAGENT.

Failure Cases. Beyond the above cases, we present additional failure cases for reference, as shown in Table 11 and Table 12. As illustrated in the tables, the majority of low-quality predictions stem from *evidence misattribution* during the reasoning process, where the model partially or fully attributes conclusions to incorrect or fabricated evidence, particularly in complex cases with dense clinical records covering multiple conditions. We attribute this failure pattern to the inherent hallucination limitations of LLMs, which become more pronounced in complex multi-condition reasoning scenarios. These findings represent an intriguing direction that warrants further systematic investigation in future work.

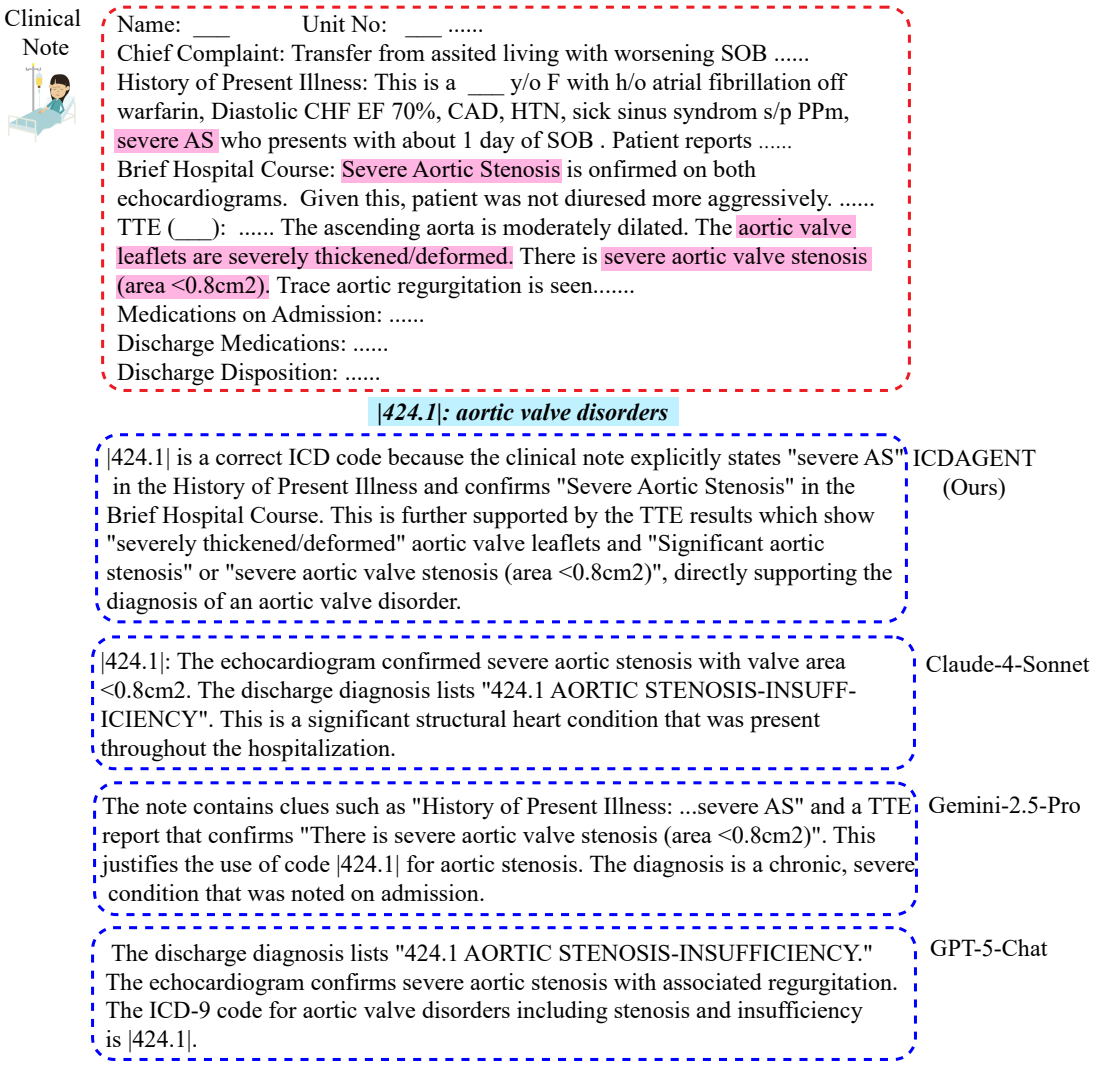

Figure 11: The explanations of "|424.1|" on the ICD-9 coding system.

Table 11: Failure Case Analysis: Evidence Misattribution (Unrelated Condition)

Category	Content
ICD Code	`|599.0|`
Explanation	`|599.0|` is a correct ICD code because the clinical note explicitly states "Urosepsis UTI" in the Brief Hospital Course and Discharge Diagnosis sections. This is supported by "positive urine cultures" which "showed her distal end on the LEFT side" and "Incidentally, a PICC which was d/c'ed on _ returned with a positive tip culture of MRSA," confirming a urinary tract infection. The note also mentions "She was began on meropenem at admission and was continued on this for 14 days for presumed urosepsis," which is treatment for a severe urinary tract infection.
LLM Judge	**Incorrect rationale.**
Reasoning Behind Judgment	The explanation correctly identifies "Urosepsis UTI" in the Brief Hospital Course and Discharge Diagnosis as primary evidence. However, the supporting evidence cited—such as "positive urine cultures" which "showed her distal end on the LEFT side" and "positive tip culture of MRSA" from a PICC line—is significantly flawed and misattributed, as these details refer to the fistula and PICC infection, not directly to a UTI. The urine culture itself was also reported as mixed flora consistent with contamination.

Table 12: Failure Case 2: Evidence Misattribution (Anatomical Region Confusion)

Category	Content
ICD Code	`|724.02|`
Explanation	`|724.02|` is a correct ICD code because the clinical note explicitly states s̈pinal stenosis L4/5ïn the Past Medical History section, which refers to spinal stenosis of the lumbar region. The MRI c-spine findings of m̈oderate bilateral C5-6 neural foraminal stenosisf̈urther support the presence of spinal stenosis, which can be associated with lower back symptoms and would be coded as lumbar stenosis in conjunction with the PMH documentation.
LLM Judge	**Partially correct rationale.**
Reasoning Behind Judgment	The code is correct for lumbar spinal stenosis based on s̈pinal stenosis L4/5ïn the PMH. However, the rationale incorrectly includes MRI findings of m̈oderate bilateral C5-6 neural foraminal stenosis(̈cervical stenosis) as supporting evidence for *lumbar* stenosis, which is a significant logical flaw.

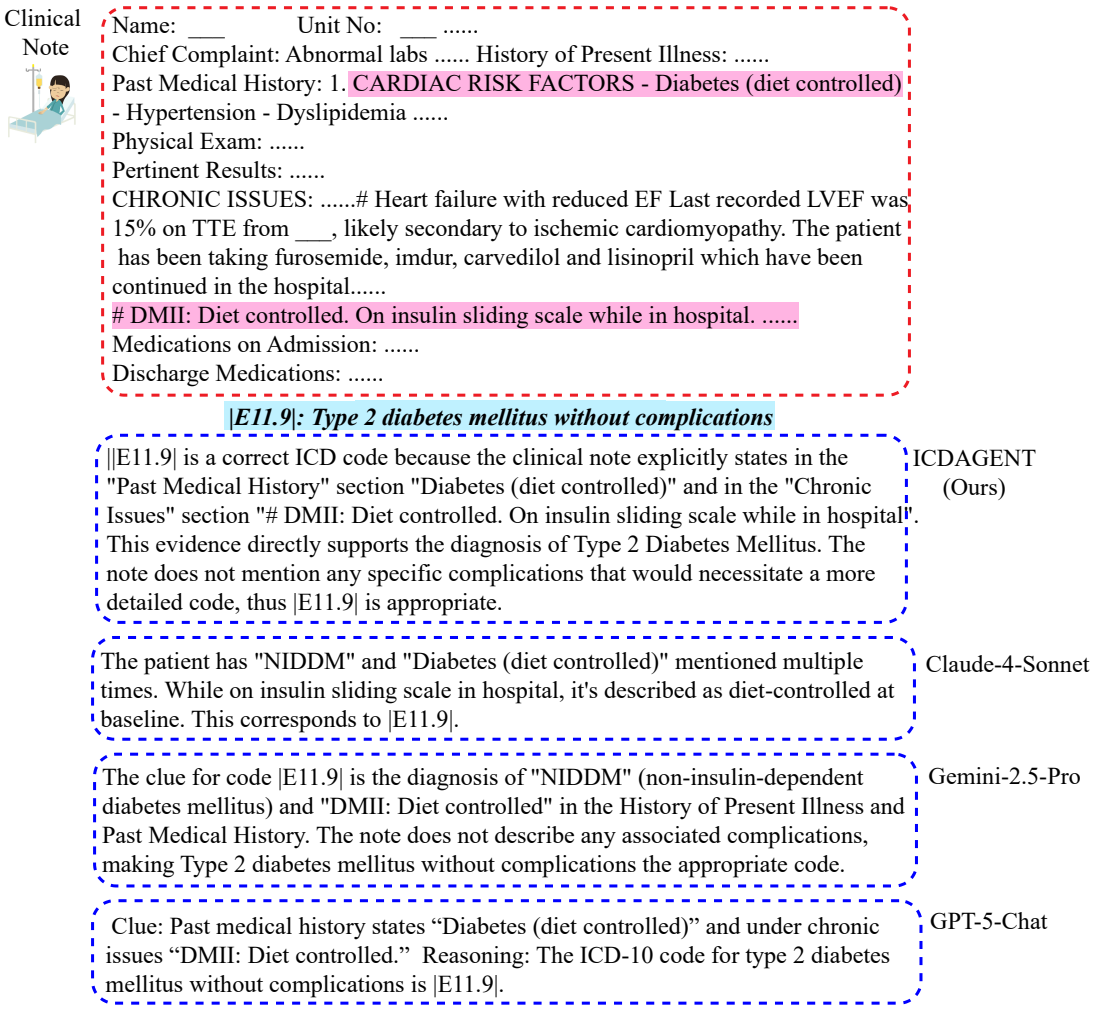

Figure 12: The explanations of “|E11.9|” on the ICD-10 coding system.