JointCoder: Exploring Automated ICD Coding on Real-World Chinese EHRs with a Multi-Agent Framework

Kangjun Liu[1,3*], Zhenyu Li[1,3*], Jianlei Wang[1,3*], Hongjiao Guan[1,3†], Ying Lian[2], Guoqiang Chen[2], Tao Xin[2], Wenpeng Lu[1,3†]

[1]Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan, China

[2]The First Affiliated Hospital of Shandong First Medical University, Shandong Provincial Qianfoshan Hospital, Jinan, China

[3]Shandong Provincial Key Laboratory of Computing Power Internet and Service Computing, Shandong Fundamental Research Center for Computer Science

lkjorigin@foxmail.com, wenpeng.lu@qlu.edu.cn

GitHub Repository Demo Video

Abstract

Automated ICD coding is a critical task for standardizing clinical information from electronic health records (EHRs) and supporting downstream healthcare administration. However, existing automated ICD coding systems face several fundamental challenges. First, the majority of existing research focuses on English ICD tasks, with limited attention to Chinese-language clinical contexts due to the scarcity of publicly available Chinese ICD datasets. Second, most approaches primarily target disease coding, overlooking procedure coding as well as the multi-stage workflows followed in real-world clinical practice. Moreover, many recent methods rely heavily on closed-source large language models or substantial computational resources, which limits their scalability and deployability in clinical environments. To address these gaps, this paper proposes *JointCoder*, which includes a real-world Chinese ICD coding dataset and a multi-agent framework that reformulates automated ICD coding as a joint disease-procedure coding task. JointCoder explicitly models real-world clinical coding workflows through stage-wise agent collaboration. All agents are instantiated using locally deployed 1.7B-parameter models, enabling scalable and privacy-preserving deployment. Extensive experiments on real-world Chinese ICD coding datasets demonstrate JointCoder's superiority over state-of-the-art baselines across all evaluation metrics.

1 Introduction

The International Classification of Diseases (ICD) is a standardized system for encoding diagnoses and health conditions, forming a critical foundation for medical billing, clinical management, and public health analytics (Yan et al., 2022; Teng et al., 2022). In current clinical practice, ICD coding remains largely a manual assignment that relies heavily on the experiences of clinical coders. It is time-consuming, labor-intensive, and error-prone, often resulting in incomplete or inaccurate code assignments that undermine reimbursement accuracy and the reliability of downstream clinical analytics (Cao et al., 2020; Dong et al., 2022; Hosseini et al., 2021). In China, this issue is particularly acute: the rapidly aging population has driven a significant increase in hospital admissions, further intensifying the burden on ICD coding (Lu et al., 2025). These limitations and pressures highlight the urgent need for accurate and efficient automated ICD coding solutions.

A series of studies on automated ICD coding have been documented in the literature, spanning from traditional machine learning-based methods to more recent deep learning and large language model (LLM)-based approaches (Shi et al., 2017; Mullenbach et al., 2018; Huang et al., 2022; Zhao et al., 2024; Liang et al., 2025a; Zhang et al., 2025b; Liang et al., 2025b). Early approaches applied feature-based classifiers to assign codes from clinical text (Medori and Fairon, 2010; Perotte et al., 2014). Subsequent deep learning methods formulate ICD coding as a multi-label text classification problem, employing convolutional or attention-based neural networks to learn representations from clinical narratives (Xie and Xing, 2018; Mullenbach et al., 2018). More recently, LLM-based approaches have demonstrated strong potential by leveraging pretrained models to directly interpret clinical text and predict codes via prompting or fine-tuning (Jaganathan et al., 2025; Li et al., 2025c).

*These authors contributed equally to this work.
†Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 263–274

Although great advancements have been achieved, most existing automated ICD coding systems face three major gaps that limit their effectiveness in real-world clinical settings. **Gap 1: Limited availability of real-world Chinese ICD datasets.** The majority of existing research focuses on English ICD tasks, while Chinese settings remain underexplored due to the scarcity of publicly available Chinese ICD datasets (Khadka et al., 2025; Yan et al., 2022; Li et al., 2025d). Given the rapidly aging population and rising hospital admissions in China, constructing high-quality Chinese ICD datasets is both necessary and urgent to advance research on automated ICD coding for Chinese healthcare. **Gap 2: Neglect of procedure coding and disease-procedure interactions.** Most existing automated ICD coding methods primarily focus on disease coding (Liang et al., 2025a; Xi et al., 2025; Vu et al., 2021; Zhang et al., 2025b; Li et al., 2025b), while procedure coding and disease-procedure interactions are largely overlooked. In real-world clinical practice, procedures provide critical complementary signals for disease interpretation and reimbursement. Ignoring their interactions with diseases leads to incomplete modeling of ICD coding workflows and suboptimal coding accuracy. **Gap 3: Heavy reliance on closed-source LLMs.** Many recent approaches depend on closed-source LLMs, resulting in high deployment costs and creating barriers to real-world adoption. These challenges include limited scalability in system maintenance and increased risks to patient privacy when transmitting EHR data to external LLM services (Lee and Lindsey, 2024; Mustafa et al., 2025; Li et al., 2025c; Liang et al., 2025b).

To bridge these gaps, we propose *JointCoder*, which comprises a real-world Chinese ICD coding dataset and a multi-agent framework for automated ICD coding that explicitly models practical clinical coding workflows. To address **Gap 1**, we construct a high-quality real-world Chinese ICD dataset derived from 6,747 EHRs collected from a top-tier Chinese hospital. JointCoder is developed and evaluated on this dataset, ensuring that the proposed multi-agent framework operates under realistic clinical documentation and coding conditions. To address **Gap 2**, JointCoder formulates automated ICD coding as a joint disease-procedure coding task and proposes a structured multi-agent framework that explicitly models real-world clinical coding workflows. Given a patient's EHR as input, the coding process is decomposed into multiple stages, such as *standardization* and *candidate mining*. Each stage is handled by a dedicated agent, and agents exchange structured intermediate outputs to capture dependencies between disease diagnoses and medical procedures, thereby enabling workflow-consistent ICD assignments. To address **Gap 3**, all agents in JointCoder are instantiated using locally deployed 1.7B-parameter models. This design avoids reliance on closed-source LLMs, reduces deployment and maintenance costs, and supports scalable and privacy-preserving inference in real-world clinical environments. Main contributions of this work are summarized below:

- We construct a high-quality real-world Chinese ICD coding dataset derived from 6,747 EHRs collected from a top-tier Chinese hospital, covering both disease codes and procedure codes for model training and evaluation.
- To the best of our knowledge, we are the first to reformulate automated ICD coding as a joint disease-procedure task that captures interactions between diseases and procedures and predicts both disease code and procedure code simultaneously.
- We propose JointCoder, a multi-agent framework built on locally deployed 1.7B-parameter models, achieving strong performance with reduced computational cost while enhancing data security and suitability for real-world clinical deployment.

2 Related Work

2.1 Automated ICD Coding

Automated ICD coding aims to assign standardized diagnosis and procedure codes from EHRs by leveraging artificial intelligence techniques, thereby improving coding efficiency and consistency (Lu et al., 2023; Wang et al., 2024; Li et al., 2025d). Existing research on automated ICD coding can be broadly categorized into three lines of work: traditional machine learning-based, deep learning-based, and LLM-based approaches. *Traditional machine learning-based approaches* rely on handcrafted features extracted from clinical text or structured data, which are then used with supervised classifiers for code assignment (Medori and Fairon, 2010; Perotte et al., 2014; Zhang et al., 2017). *Deep learning-based approaches* typically formulate ICD coding as a multi-label text classification problem and employ neural architectures to

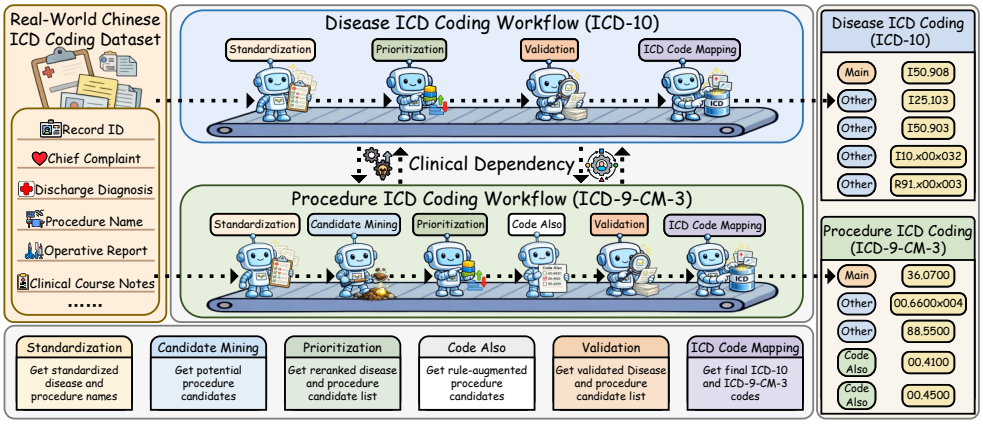

Figure 1: Overview of the real-world Chinese ICD coding dataset and the JointCoder framework. JointCoder jointly performs disease coding (ICD-10) and procedure coding (ICD-9-CM-3) through a unified, workflow-aligned multi-stage multi-agent framework.

learn representations directly from clinical narratives (Wu et al., 2024b,a; Liang et al., 2025a). *LLM-based approaches* leverage pretrained large language models to enhance ICD code prediction through improved semantic understanding, reasoning ability, and generalization across clinical contexts (Motzfeldt et al., 2025; Barreiros et al., 2025).

Despite their effectiveness, most existing methods primarily emphasize model architectures, while paying limited attention to alignment with real-world clinical coding workflows.

2.2 Collaborative Multi-Agent Reasoning

Recent advances in LLMs have spurred growing interest in collaborative multi-agent frameworks, which enhance reasoning performance through coordination and interaction among multiple agents (Li et al., 2024; Chen et al., 2024b; Tran et al., 2025). Existing research in this area can be broadly categorized into role-based collaboration, debate-based collaboration, and domain-specific multi-agent systems. *Role-based approaches* assign distinct functional roles to agents to support complementary reasoning and task decomposition (Chen et al., 2024a; Li et al., 2025a; Zong et al., 2024). *Debate-based approaches* improve prediction quality through structured interactions, critiques, and iterative refinement among agents (Feng et al., 2025; Zhang et al., 2025a; Zhao et al., 2025). *Domain-specific systems* tailor multi-agent collaboration to specialized application settings by incorporating domain knowledge and task-specific constraints (Fan et al., 2025; Wang et al., 2025).

Building on these advances, this work focuses on leveraging a multi-agent framework that explicitly models real-world clinical coding workflows followed by professional coders.

3 Methodology

As shown in Figure 1, JointCoder consists of the Chinese ICD coding dataset and the multi-agent framework for joint disease-procedure ICD coding. The figure provides a schematic overview of the data pipeline and the multi-agent collaboration process. The details of each component are described in the following subsections.

3.1 Real-World Chinese ICD Coding Dataset

We construct a real-world dataset for automated ICD coding based on patient EHRs collected from a top-tier Chinese hospital. The dataset comprises 6,747 hospitalization records, primarily covering cardiology-related clinical encounters. Each record corresponds to a single inpatient stay and contains comprehensive clinical information, including *Record ID*, *Chief Complaint*, *Discharge Diagnosis*, *Procedure Name*, *Operative Report*, *Clinical Course Notes*, and other relevant clinical fields.

All records are manually annotated by experienced coders in accordance with standardized coding guidelines, assigning ICD-10 disease codes (World Health Organization, 2004) and ICD-9-CM-3 procedure codes (National Center for Health Statistics, 1980). In addition, the data are rigorously cleaned and de-identified to ensure high quality and protect patient privacy. Construction details are provided in Appendix A.

3.2 JointCoder Architecture

JointCoder is a multi-agent framework for automated ICD coding that explicitly models real-world clinical coding workflows. As illustrated in Figure 1, given an EHR as input, JointCoder consists of six stages: *Standardization*, *Candidate Min-*

ing, *Prioritization*, *Code Also*, *Validation*, and *ICD Code Mapping*. Disease coding utilizes a subset of these stages, while procedure coding follows the complete pipeline. Table 5 presents example outputs at each stage. JointCoder is implemented using seven locally deployed Qwen3-1.7B (Yang et al., 2025) agents, each fine-tuned for a specific subtask. Details of each stage are described below.

Standardization. In real-world clinical coding, disease coding primarily relies on the structured *Discharge Diagnosis* field in the EHR, while procedure coding depends on the structured *Procedure Name* field, which summarizes the patient's main surgical interventions. However, these fields often contain non-standard expressions, including physician-specific writing styles, abbreviations, colloquial terms, and concatenated mentions of multiple conditions or procedures, making them difficult to directly map to ICD codes.

To address this issue, JointCoder employs dedicated standardization agents to normalize raw diagnosis and procedure descriptions into canonical ICD terminology by leveraging both the target field and the full EHR context. Formally, given a discharge diagnosis text D and a procedure name text S with their associated EHR, we first extract raw diagnosis and procedure sets $\{d_1, \ldots, d_n\}$ and $\{s_1, \ldots, s_m\}$. The agent then transforms them into standardized sets $\{d_1^{\text{std}}, \ldots, d_n^{\text{std}}\}$ and $\{s_1^{\text{std}}, \ldots, s_m^{\text{std}}\}$, where each d_i^{std} and s_i^{std} is mapped to a canonical ICD-10 disease name and ICD-9-CM-3 procedure name, respectively.

Candidate Mining. In real-world ICD coding, relying solely on the *Procedure Name* field is often insufficient, as auxiliary operations required by coding standards may be omitted (with 63.8% missing in our investigations). Therefore, JointCoder further mines potential procedures from the full EHR. In contrast, the *Discharge Diagnosis* field provides near-complete coverage for disease coding (with only 0.8% missing), and the remaining omissions are addressed during the *Validation* stage. Hence, this stage primarily targets procedure coding.

To this end, JointCoder employs a dedicated procedure mining agent, implemented with a locally fine-tuned model, to identify omitted auxiliary operations from the full EHR. Given the standardized procedure list $\{s_1^{\text{std}}, \ldots, s_m^{\text{std}}\}$ produced in *Standardization* and the associated EHR, the agent extracts additional candidate procedures missing from the *Procedure Name* field. The standardized procedures and the newly identified candidates are then merged to form a candidate set $\{s_1^{\text{cand}}, \ldots, s_k^{\text{cand}}\}$ for subsequent prioritization and validation.

Prioritization. ICD coding requires diagnosis and procedure codes to be ordered by clinical importance, with the principal diagnosis or procedure listed first. However, previous stages produce unordered candidate sets, necessitating an explicit prioritization step to ensure clinically consistent ranking. This step is particularly critical for procedure coding, as the subsequent *Code Also* expansion depends on the principal procedure.

JointCoder therefore employs dedicated prioritization agents to rank diagnosis and procedure candidates under the full EHR context. Formally, given candidate sets $\{d_1^{\text{std}}, \ldots, d_n^{\text{std}}\}$ and $\{s_1^{\text{cand}}, \ldots, s_k^{\text{cand}}\}$ with the associated EHR, the agents output ordered lists $\{d_1^{\text{rank}}, \ldots, d_n^{\text{rank}}\}$ and $\{s_1^{\text{rank}}, \ldots, s_k^{\text{rank}}\}$, in which the first item is treated as the main diagnosis or procedure, respectively.

Code Also. In ICD-9-CM-3 procedure coding, after identifying the principal procedure in *Prioritization*, an additional *Code Also* step is often required to supplement it with clinically relevant accompanying operations that must be coded separately. For instance, stent implantation may require extra codes to specify the number of vessels treated or stents inserted.

To support this, JointCoder builds an ICD-9-CM-3 *Code Also* rule index using Elasticsearch for retrieval.[1] Given the ranked procedure list $\{s_1^{\text{rank}}, \ldots, s_k^{\text{rank}}\}$, the system retrieves *Code Also* entries based on the principal procedure and appends them to form an expanded procedure set $\{s_1^{\text{extra}}, \ldots, s_t^{\text{extra}}\}$. Any redundant entries are subsequently filtered out during the *Validation* stage.

Validation. The preceding stages may introduce common coding errors, including over-coding, under-coding, and mis-coding. This issue is particularly pronounced for procedure coding after the *Code Also* step, which can introduce redundant candidates. Therefore, a validation step is required to align candidate codes with EHR evidence and ensure coding consistency.

JointCoder employs dedicated validation agents to verify diagnosis and procedure candidates against the full EHR context. Formally, given candidate sets $\{d_1^{\text{rank}}, \ldots, d_n^{\text{rank}}\}$ and $\{s_1^{\text{extra}}, \ldots, s_t^{\text{extra}}\}$ with the associated EHR, the agents perform

[1] https://www.elastic.co.

Setting		Disease (ICD-10)				Procedure (ICD-9-CM-3)			
Method	Model	Precision	Recall	F1-score	Jaccard	Precision	Recall	F1-score	Jaccard
Vanilla	Qwen3-1.7B	0.5842	0.4105	0.4822	0.3352	0.2222	0.1469	0.1769	0.1721
	Qwen3-4B	0.6236	0.4888	0.5480	0.3893	0.2007	0.1828	0.1913	0.1862
	Qwen3-8B	0.6168	0.5661	0.5904	0.4144	0.1930	0.1641	0.1774	0.1811
	Qwen3-14B	0.5787	0.5102	0.5423	0.3792	0.1688	0.2266	0.1935	0.1927
	DeepSeek-V3.2†	0.4039	0.5109	0.4512	0.2953	0.0945	0.2516	0.1374	0.1916
	GPT-5-2025-08-07†	0.4205	0.4367	0.4284	0.2761	0.0620	0.1328	0.0846	0.0720
Few-shot	Qwen3-1.7B	0.4405	0.3687	0.4015	0.2496	0.2945	0.2328	0.2600	0.2079
	Qwen3-4B	0.5440	0.5121	0.5276	0.3641	0.3421	0.2438	0.2847	0.2305
	Qwen3-8B	0.4986	0.5461	0.5213	0.3484	0.2049	0.1578	0.1783	0.1250
	Qwen3-14B	0.5553	0.5738	0.5644	0.4005	0.3688	0.2922	0.3261	0.3115
	DeepSeek-V3.2†	0.4926	0.5282	0.5098	0.3430	0.1791	0.2547	0.2103	0.2623
	GPT-5-2025-08-07†	0.4378	0.4829	0.4593	0.3014	0.1041	0.1469	0.1218	0.1364
CoT	Qwen3-1.7B	0.5965	0.4332	0.5019	0.3379	0.2956	0.1797	0.2235	0.1965
	Qwen3-4B	0.6565	0.3368	0.4452	0.2782	0.3373	0.2203	0.2665	0.2151
	Qwen3-8B	0.6343	0.4216	0.5065	0.3229	0.3867	0.2641	0.3138	0.2735
	Qwen3-14B	0.6475	0.3360	0.4424	0.2943	0.3603	0.2297	0.2805	0.2554
	DeepSeek-V3.2†	0.3896	0.4866	0.4327	0.2816	0.3114	0.2438	0.2734	0.3190
	GPT-5-2025-08-07†	0.5211	0.4763	0.4977	0.3239	0.1102	0.0859	0.0966	0.1027
CoT-SC	Qwen3-1.7B	0.5662	0.3743	0.4507	0.3030	0.3683	0.1922	0.2526	0.2296
	Qwen3-4B	0.6264	0.4748	0.5402	0.3829	0.3656	0.2359	0.2868	0.2407
	Qwen3-8B	0.6135	0.5317	0.5697	0.3959	0.3536	0.2453	0.2897	0.2476
	Qwen3-14B	0.5411	0.3802	0.4466	0.3022	0.3916	0.2766	0.3242	0.2959
	DeepSeek-V3.2†	0.5016	0.5040	0.5028	0.3315	0.6499	0.3625	0.4654	0.4805
	GPT-5-2025-08-07†	0.7357	0.3859	0.5063	0.3260	0.3585	0.2375	0.2857	0.2621
AiCoder	Qwen3-1.7B	0.6405	0.5384	0.5851	0.4134	0.5857	0.4815	0.5286	0.3592
	Qwen3-4B	0.6628	0.5523	0.6025	0.4311	0.6023	0.4927	0.5422	0.3717
	Qwen3-8B	0.6812	0.5622	0.6160	0.4451	0.6159	0.5031	0.5538	0.3829
	Qwen3-14B	0.7056	0.5723	0.6321	0.4620	0.6314	0.5121	0.5657	0.3942
	DeepSeek-V3.2†	0.7124	0.5901	0.6455	0.4766	0.6387	0.5192	0.5728	0.4013
	GPT-5-2025-08-07†	0.7482	0.5607	0.6412	0.4717	0.6552	0.5028	0.5691	0.3976
SFT	Qwen3-1.7B	0.8241	0.8262	0.8252	0.7349	0.8022	0.8026	0.7840	0.7828
	Qwen3-4B	0.8366	0.8435	0.8401	0.7500	0.8268	0.7587	0.7867	0.7877
	Qwen3-8B	0.8496	0.8532	0.8514	0.7633	0.8296	0.7531	0.7895	0.7909
	Qwen3-14B	0.8551	0.8646	0.8599	0.7730	0.8313	0.7625	0.7954	0.7762
JointCoder	Qwen3-1.7B	**0.9067***	**0.9122***	**0.9094***	**0.8547***	**0.8554***	**0.8676***	**0.8615***	**0.8618***

Table 1: Performance of different methods on the proposed ICD coding dataset. Models marked with † denote commercial APIs. The best and second-best results are highlighted in **bold** and underline, respectively. Results marked with * indicate statistically significant improvements over the strongest baseline ($p < 0.05$).

evidence-based validation to remove unsupported codes, recover omissions, and correct errors, producing refined and finalized diagnosis and procedure lists $\{d_1^{\text{val}}, \ldots, d_r^{\text{val}}\}$ and $\{s_1^{\text{val}}, \ldots, s_u^{\text{val}}\}$.

ICD Code Mapping. After the preceding stages, JointCoder produces validated disease and procedure names rather than ICD codes, necessitating an explicit mapping step. To support this, we construct an ICD index using Elasticsearch, which stores the complete set of official ICD-10 disease codes and ICD-9-CM-3 procedure codes along with their standardized names. The index preserves the hierarchical structure of the ICD system to maintain parent-child relationships.

JointCoder then performs retrieval-based mapping by querying the index. Formally, given finalized name lists $\{d_1^{\text{val}}, \ldots, d_r^{\text{val}}\}$ and $\{s_1^{\text{val}}, \ldots, s_u^{\text{val}}\}$, the system retrieves the corresponding codes for each validated name, respectively, producing the final ICD codes for diseases and procedures.

4 Experimental Setting

We conduct extensive experiments to evaluate the performance of JointCoder by answering the following key research questions:

- **RQ1**: Does JointCoder outperform LLMs of different scales?
- **RQ2**: Does JointCoder achieve better performance than fine-tuned LLMs?
- **RQ3**: Does JointCoder outperform state-of-the-art ICD coding methods?

4.1 Evaluation Metrics

We evaluate JointCoder on the multi-label ICD coding task using *Accuracy*, *Precision*, *Recall*, and *F1-score*. To capture prediction quality at the set level, we further report *Jaccard* similarity (Jaccard, 1901), which reflects the overlap between the predicted and ground-truth ICD code sets.

4.2 Baselines and Implementation Details

We compare JointCoder with state-of-the-art ICD coding methods and strong LLM baselines. Specifically, we include general-purpose LLM prompting strategies, including Vanilla (Zero-shot), Few-shot (Brown et al., 2020), Chain-of-Thought (CoT) (Wei et al., 2022), and CoT with Self-Consistency (CoT-SC) (Wang et al., 2023), as well as supervised fine-tuned LLMs. We further consider AiCoder (Liang et al., 2025b), a recently pro-

posed multi-agent framework for automated ICD coding, as a strong multi-agent baseline.

All methods are evaluated under a unified experimental setup. The dataset is split into training and test sets with a 9:1 ratio. Models are trained for 5 epochs with a learning rate of $1e^{-5}$ and a maximum input length of 8192 tokens. All experiments are conducted on a server equipped with eight NVIDIA RTX 4090 GPUs running Ubuntu 22.04.4 LTS. Additional analyses, including case studies, efficiency evaluation, and ablation experiments, are provided in Appendices B–D.

5 Result Analysis

5.1 Comparison with LLM Baselines (RQ1)

Table 1 reports the performance of JointCoder and LLM baselines under different prompting settings. JointCoder consistently outperforms all prompting-based methods.

Vanilla and Few-shot. Vanilla and Few-shot prompting exhibit limited performance across model sizes, particularly for procedure coding, where both *Precision* and *Recall* remain low. This suggests that directly prompting LLMs to generate ICD codes without structured guidance is insufficient for handling the complexity of real-world coding. In contrast, JointCoder achieves substantially higher *F1-scores* on both tasks, despite using a smaller backbone model.

CoT and CoT-SC. COT and CoT-SC improve *Recall* by encouraging step-by-step reasoning. However, this often comes at the cost of reduced *Precision*, especially for procedure coding, due to over-generation of auxiliary or irrelevant procedures. Consequently, their overall *F1-score* and *Jaccard* scores remain significantly lower than those of JointCoder. This indicates that generic reasoning prompts alone are insufficient without structured modeling of clinical coding workflows.

5.2 Comparison with Fine-Tuned LLMs (RQ2)

To assess whether JointCoder's performance gain can be attributed merely to supervised fine-tuning, we compare it with LLMs fine-tuned on the same training data, including Qwen3 models ranging from 1.7B to 14B parameters. As shown in Table 1, fine-tuning substantially improves ICD coding performance. However, fine-tuned models still lag behind JointCoder by approximately 5~6% in both *F1-score* and *Jaccard* on average.

Furthermore, increasing model size yields only limited improvements for the fine-tuned models, while JointCoder achieves the best performance with a smaller backbone. This suggests that simply scaling up or fine-tuning a model is not sufficient for ICD coding. In contrast, explicitly modeling the clinical coding process plays a more important role in improving performance.

5.3 Comparison with State-of-the-Art ICD Coding Methods (RQ3)

We compare JointCoder with AiCoder (Liang et al., 2025b), a recent multi-agent framework for automated ICD coding that incorporates knowledge-graph-enhanced re-ranking and retrieval mechanisms to support hierarchical code assignment.

As shown in Table 1, AiCoder achieves *F1-scores* of 0.6455 on ICD-10 and 0.5728 on ICD-9-CM-3. In contrast, JointCoder attains 0.9094 and 0.8615, respectively, consistently outperforming AiCoder across *Precision*, *Recall*, *F1-score*, and *Jaccard*. These results highlight the advantage of JointCoder in explicitly modeling real-world clinical coding workflows for ICD code prediction.

6 Conclusion

Automated ICD coding remains a challenging problem, as existing approaches often rely on generic prompting strategies or monolithic modeling paradigms that fail to reflect real-world clinical coding workflows. To support realistic evaluation, we construct a real-world Chinese ICD dataset derived from cardiology electronic health records. Building upon this dataset, we introduce JointCoder, a multi-agent framework for automated ICD coding that aligns model reasoning with the structured stages of clinical coding practice. By decomposing ICD coding into structured stages and jointly modeling disease and procedure prediction, JointCoder produces more consistent and clinically aligned coding results. Extensive experiments demonstrate the effectiveness of JointCoder in substantially improving ICD coding performance across multiple settings. A key takeaway is that modeling real-world clinical coding workflows is crucial for reliable automated ICD coding. In future work, extending this workflow-based approach to other specialties and more complex clinical records may further improve its applicability.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62376130, No.72501151), Program of New Twenty Policies for Universities of Jinan (No.202333008), the Open Project of the Key Laboratory of Computing Power Network and Information Security, Ministry of Education (No. 2024ZD017), the Pilot Project for Integrated Innovation of Science, Education, and Industry of Qilu University of Technology (Shandong Academy of Sciences) (No.2025ZDZX01), and Shandong Talent Introduction Program (No.WSR2025005).

Limitations

While JointCoder demonstrates strong performance on real-world ICD coding tasks, several limitations remain.

First, the proposed dataset and experimental evaluation are currently restricted to cardiology-related clinical encounters. Although this setting reflects realistic clinical practice, broader validation across multiple medical specialties and real hospital deployments is necessary to assess generalizability and practical impact.

Second, both the dataset and the JointCoder framework are developed specifically for Chinese EHRs. While this allows accurate modeling of Chinese clinical documentation and coding conventions, adapting the approach to other languages would require additional preprocessing and adjustments to language-specific documentation styles and coding practices.

Third, despite reducing common ICD coding errors through workflow-based multi-stage reasoning, JointCoder is built upon large language models and thus remains susceptible to hallucinations. In high-stakes domains such as healthcare, factual inaccuracies can pose significant risks. Addressing hallucinations in LLM-based clinical systems remains an important direction for future research.

Ethics Statement

We acknowledge that all authors of this work are aware of and comply with the ACL Code of Ethics. All datasets and derived artifacts are used strictly for research purposes. The dataset is constructed from real-world EHRs that have been rigorously de-identified and anonymized to protect patient privacy and data security.

This work advances automated ICD coding by aligning LLM-based agent reasoning with real-world clinical coding workflows, aiming to support rather than replace professional clinical coders. Automated ICD coding is inherently high-stakes, as errors may affect downstream processes such as clinical analytics, insurance reimbursement, and healthcare management. Although JointCoder improves coding reliability, rigorous validation, continuous monitoring, and alignment with institutional governance are essential prior to real-world deployment. We envision this work as a step toward the responsible and transparent integration of large language models into safety-critical healthcare systems. In addition, this study was reviewed and approved by the Ethics Committee of the First Affiliated Hospital of Shandong First Medical University.

References

Leonor Barreiros, Isabel Coutinho, Gonçalo Correia, and Bruno Martins. 2025. Explainable ICD coding via entity linking. In *Proceedings of the 2nd Workshop on Patient-Oriented Language Processing*, pages 219–227.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Proceedings of the 34th Conference on Neural Information Processing Systems*, pages 1877–1901.

Pengfei Cao, Chenwei Yan, Xiangling Fu, Yubo Chen, Kang Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. 2020. Clinical-Coder: Assigning interpretable ICD-10 codes to chinese clinical notes. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 294–301.

Pei Chen, Shuai Zhang, and Boran Han. 2024a. CoMM: Collaborative multi-agent, multi-reasoning-path prompting for complex problem solving. In *Proceedings of the 17th North American Chapter of the Association for Computational Linguistics*, pages 1720–1738.

Shuaihang Chen, Yuanxing Liu, Wei Han, Weinan

Zhang, and Ting Liu. 2024b. A survey on LLM-based multi-agent system: Recent advances and new frontiers in application. *arXiv preprint arXiv:2412.17481*.

Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, and 1 others. 2025. PaddleOCR 3.0 technical report. *arXiv preprint arXiv:2507.05595*.

Hang Dong, Matúš Falis, William Whiteley, Beatrice Alex, Joshua Matterson, Shaoxiong Ji, Jiaoyan Chen, and Honghan Wu. 2022. Automated clinical coding: what, why, and where we are? *NPJ digital medicine*, 5(1):159.

Zhihao Fan, Lai Wei, Jialong Tang, Wei Chen, Wang Siyuan, Zhongyu Wei, and Fei Huang. 2025. AI hospital: Benchmarking large language models in a multi-agent medical interaction simulator. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10183–10213.

Zhaopeng Feng, Jiayuan Su, Jiamei Zheng, Jiahan Ren, Yan Zhang, Jian Wu, Hongwei Wang, and Zuozhu Liu. 2025. M-MAD: Multidimensional multi-agent debate for advanced machine translation evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pages 7084–7107.

Nafiseh Hosseini, Khalil Kimiafar, Sayyed Mostafa Mostafavi, Behzad Kiani, Kazem Zendehdel, Armin Zareiyan, and Saeid Eslami. 2021. Factors affecting the quality of diagnosis coding data with a triangulation view: A qualitative study. *The International Journal of Health Planning and Management*, 36(5):1666–1684.

Chao-Wei Huang, Shang-Chi Tsai, and Yun-Nung Chen. 2022. PLM-ICD: Automatic ICD coding with pretrained language models. In *Proceedings of the 4th Clinical Natural Language Processing Workshop*, pages 10–20.

Paul Jaccard. 1901. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37(1):547–579.

Guna Sekaran Jaganathan, Indika Kahanda, and Upulee Kanewala. 2025. Metamorphic testing for robustness and fairness evaluation of LLM-based automated ICD coding applications. *Smart Health*, 36(1):100564.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20(4):422–446.

Supriya Khadka, Xiaorui Jiang, and Vasile Palade. 2025. Data quality in clinical coding: A critical analysis and preliminary study. *medRxiv*, 1(1):2025–08.

Simon A Lee and Timothy Lindsey. 2024. Can large language models abstract medical coded language? *arXiv preprint arXiv:2403.10822*.

Haoran Li, Ziyi Su, Yun Xue, Zhiliang Tian, Yiping Song, and Minlie Huang. 2025a. Advancing collaborative debates with role differentiation through multi-agent reinforcement learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pages 22655–22666.

Mingyang Li, Viktor Schlegel, Tingting Mu, Warren Del-Pinto, and Goran Nenadic. 2025b. Structured information matters: Explainable ICD coding with patient-level knowledge graphs. *arXiv preprint arXiv:2509.09699*, pages 1–16.

Rumeng Li, Xun Wang, and Hong Yu. 2025c. Improving rare and common ICD coding via a multi-agent LLM-based approach. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 4945–4949.

Xiaobo Li, Yijia Zhang, Xiaodi Hou, Shilong Wang, and Hongfei Lin. 2025d. Deep learning for automatic ICD coding: Review, opportunities and challenges. *Artificial Intelligence in Medicine*, 168(1):103187.

Xinyi Li, Sai Wang, Siqi Zeng, Yu Wu, and Yi Yang. 2024. A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 1(1):9.

Zhenpeng Liang, Hongjiao Guan, Wenpeng Lu, Xueping Peng, Bing Xu, and Muyun Yang. 2025a. HiRes: Hierarchical feature optimization and rescorer for automatic ICD coding. In

Proceedings of the 50th IEEE International Conference on Acoustics, Speech, and Signal Processing, pages 1–5.

Zhenpeng Liang, Hongjiao Guan, Weiyu Zhang, Ying Lian, Bing Xu, and Wenpeng Lu. 2025b. AiCoder: Exploring automated ICD coding on chinese EMRs with a multi-agent framework. In *Proceedings of the 19th IEEE International Conference on Bioinformatics and Biomedicine*, pages 3823–3827.

Chang Lu, Chandan Reddy, Ping Wang, and Yue Ning. 2023. Towards semi-structured automatic ICD coding via tree-based contrastive learning. In *Proceedings of the 37th Conference on Neural Information Processing Systems*, pages 68300–68315.

Wenpeng Lu, Kangjun Liu, Jianlei Wang, Xueping Peng, Tao Shen, Fa Zhu, Weiyu Zhang, Jiabing Zhu, Tao Xin, and Athanasios V Vasilakos. 2025. Advancing Chinese conversation-based patient guidance with a benchmark and knowledge-evolvable assistant. *IEEE Journal of Biomedical and Health Informatics*, 1(1):1–12.

Julia Medori and Cédrick Fairon. 2010. Machine learning and features selection for semi-automatic ICD-9-CM encoding. In *Proceedings of the 11th Annual Meeting of the North American chapter of the Association for Computational Linguistics*, pages 84–89.

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. In *Proceedings of the 30th Conference on Empirical Methods in Natural Language Processing*, pages 22612–22627.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. In *Proceedings of the 16th North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1101–1111.

Akram Mustafa, Usman Naseem, and Mostafa Rahimi Azghadi. 2025. Large language models vs human for classifying clinical documents. *International Journal of Medical Informatics*, 195(1):105800.

National Center for Health Statistics. 1980. *International Classification of Diseases, Ninth Revision, Clinical Modification: Procedures: Tabular List and Alphabetic Index*. U.S. Department of Health and Human Services.

Adler Perotte, Rimma Pivovarov, Karthik Natarajan, Nicole Weiskopf, Frank Wood, and Noémie Elhadad. 2014. Diagnosis code assignment: models and evaluation metrics. *Journal of the American Medical Informatics Association*, 21(2):231–237.

Haoran Shi, Pengtao Xie, Zhiting Hu, Ming Zhang, and Eric P Xing. 2017. Towards automated ICD coding using deep learning. *arXiv preprint arXiv:1711.04075*.

Fei Teng, Yiming Liu, Tianrui Li, Yi Zhang, Shuangqing Li, and Yue Zhao. 2022. A review on deep neural networks for ICD coding. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4357–4375.

Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D Nguyen. 2025. Multi-agent collaboration mechanisms: A survey of LLMs. *arXiv preprint arXiv:2501.06322*.

Thanh Vu, Dat Quoc Nguyen, and Anthony Nguyen. 2021. A label attention model for ICD coding from clinical text. In *Proceedings of the 29th International Conference on International Joint Conferences on Artificial Intelligence*, pages 3335–3341.

Rui Wang, Yonghe Chen, Weiyu Zhang, Jiasheng Si, Hongjiao Guan, Xueping Peng, and Wenpeng Lu. 2025. MedConMA: A confidence-driven multi-agent framework for medical Q&A. In *Proceedings of the 29th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 421–433.

Xindi Wang, Robert Mercer, and Frank Rudzicz. 2024. Multi-stage retrieve and re-rank model for automatic medical coding recommendation. In *Proceedings of the 17th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4881–4891.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang,

Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *Proceedings of the 11th International Conference on Learning Representations*, pages 1–24.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th Conference on Neural Information Processing Systems*, pages 24824–24837.

World Health Organization. 2004. *ICD-10: International statistical classification of diseases and related health problems: 10th revision*. World Health Organization.

John Wu, David Wu, and Jimeng Sun. 2024a. Beyond label attention: Transparency in language models for automated medical coding via dictionary learning. In *Proceedings of the 29th Conference on Empirical Methods in Natural Language Processing*, pages 8848–8871.

Yuzhou Wu, Xuechen Chen, Xin Yao, Yongang Yu, and Zhigang Chen. 2024b. Hyperbolic graph convolutional neural network with contrastive learning for automated ICD coding. *Computers in Biology and Medicine*, 168(1):107797.

Suyang Xi, Jiesen Shi, Jiachen Yan, MingJing Lin, Xinyi Zhou, Yuan Cheng, Hong Ding, and Chia Chao Kang. 2025. Breaking barriers in ICD classification with a robust graph neural network for hierarchical coding. *Scientific Reports*, 15(1):25676.

Pengtao Xie and Eric Xing. 2018. A neural architecture for automated ICD coding. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1066–1076.

Chenwei Yan, Xiangling Fu, Xien Liu, Yuanqiu Zhang, Yue Gao, Ji Wu, and Qiang Li. 2022. A survey of automated international classification of diseases coding: Development, challenges, and applications. *Intelligent Medicine*, 2(3):161–173.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Danchen Zhang, Daqing He, Sanqiang Zhao, and Lei Li. 2017. Enhancing automatic ICD-9-CM code assignment for medical texts with pubmed. In *Proceedings of the 16th Biomedical Natural Language Processing Workshop*, pages 263–271.

Kaiyuan Zhang, Qian Liu, Luyang Zhang, Chaoqun Zheng, Shuaimin Li, Bing Xu, Muyun Yang, Xinxiao Qiao, and Wenpeng Lu. 2025a. MADAWSD: Multi-agent debate framework for adversarial word sense disambiguation. In *Proceedings of the 30th Conference on Empirical Methods in Natural Language Processing*, pages 22294–22313.

Xu Zhang, Kun Zhang, Wenxin Ma, Rongsheng Wang, Chenxu Wu, Yingtai Li, and S Kevin Zhou. 2025b. A general knowledge injection framework for ICD coding. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pages 7180–7189.

Xueguan Zhao, Wenpeng Lu, Chaoqun Zheng, Weiyu Zhang, Jiasheng Si, and Deyu Zhou. 2025. Plan dynamically, express rhetorically: A debate-driven rhetorical framework for argumentative writing. In *Proceedings of the 30th Conference on Empirical Methods in Natural Language Processing*, pages 9562–9584.

Zhizhuo Zhao, Wenpeng Lu, Xueping Peng, Lumin Xing, Weiyu Zhang, and Chaoqun Zheng. 2024. Automated ICD coding via contrastive learning with back-reference and synonym knowledge for smart self-diagnosis applications. *IEEE Transactions on Consumer Electronics*, 70(3):6042–6053.

Chang Zong, Yuchen Yan, Weiming Lu, Jian Shao, Yongfeng Huang, Heng Chang, and Yueting Zhuang. 2024. Triad: A framework leveraging a multi-role LLM-based agent to solve knowledge base question answering. In *Proceedings of the 29th Conference on Empirical Methods in Natural Language Processing*, pages 1698–1710.

A Dataset Construction Details

The dataset was retrospectively constructed from consecutive inpatient EHRs with ethics approval, following standardized preprocessing and de-identification procedures. ICD codes were assigned by certified hospital coders in accordance with national ICD-10 and ICD-9-CM-3 guidelines, and a subset of records was independently audited to ensure annotation consistency. The code distribution exhibits a typical long-tailed pattern, reflecting real-world clinical practice.

B Case Study and Error Analysis

To further examine model behavior in a realistic setting, we present a representative case study in Table 4, where green codes denote correct predictions and red codes indicate errors.

For disease coding, prompting-based baselines are able to identify several major diagnoses but frequently confuse closely related variants, introduce unsupported codes, or miss clinically required conditions. Few-shot prompting often amplifies these issues by generating additional irrelevant diagnoses. Although supervised fine-tuning improves coverage, it still produces incorrect variants and fails to maintain clinically consistent code ordering.

Procedure coding errors are more severe. Most baselines under-predict the required procedure set, often returning only a small subset while missing essential auxiliary codes. They also struggle with fine-grained distinctions and occasionally generate unrelated or incorrectly ordered procedures. While fine-tuning slightly alleviates under-prediction, errors in granularity and ordering remain.

From this case study, we identify three dominant types of errors, including incorrect code selection, omission of clinically required codes, and inconsistent ordering. In contrast, JointCoder exactly matches the ground truth for both diseases and procedures, producing the complete code set in the correct order. This improvement stems from its workflow-based multi-stage reasoning, which promotes standardized prediction, improves completeness, and filters unsupported outputs.

C Computational Efficiency

We evaluate the efficiency of JointCoder under a practical deployment setting. The complete system, including all agent modules, OCR services, and the web interface, requires approximately 25 GB of GPU memory.

Model	Pre.	Rec.	F1.	Jac.	NDCG@3
JointCoder	**0.9067**	**0.9122**	**0.9094**	**0.8547**	**0.9190**
w/o Standardization	0.8861	0.8563	0.8710	0.7963	0.8791
w/o Prioritization	0.9008	0.8833	0.8939	0.8311	0.8266
w/o Validation	0.8912	0.8791	0.8898	0.8260	0.8844

Table 2: Ablation results for disease ICD coding.

Model	Pre.	Rec.	F1	Jac.	NDCG@3
JointCoder	**0.8554**	**0.8677**	**0.8615**	**0.8313**	**0.9018**
w/o Standardization	0.2716	0.2482	0.2594	0.1939	0.2755
w/o Candidate Mining	0.8470	0.8315	0.8423	0.8091	0.8863
w/o Prioritization	0.8474	0.8346	0.8402	0.8058	0.8167
w/o Code Also	0.7471	0.5078	0.6047	0.5671	0.8657
w/o Validation	0.3679	0.7815	0.5003	0.4167	0.5546

Table 3: Ablation results for procedure ICD coding.

For PDF-formatted clinical records, JointCoder processes each EHR in 15~21 seconds, covering OCR parsing, multi-stage reasoning, and code mapping. Notably, OCR accounts for roughly 90% of the total processing time. When structured clinical text is directly available, average processing time decreases to approximately 1.8 seconds per EHR.

These results indicate that JointCoder achieves efficient ICD coding with moderate computational overhead, supporting secure and privacy-preserving on-premise deployment in real-world clinical environments.

D Ablation Study

We conduct ablation studies by removing individual stages of JointCoder to assess their contributions. As shown in Table 2 and Table 3, removing any stage leads to clear performance degradation, with more pronounced drops in procedure coding.

To further evaluate the quality of clinically prioritized ordering, we report NDCG@3 (Järvelin and Kekäläinen, 2002). JointCoder consistently outperforms its ablated variants under this metric, indicating the importance of each stage for maintaining accurate and well-ordered predictions.

E Web-based System Interface

To facilitate real-world deployment, we develop a web-based system interface for JointCoder that supports end-to-end ICD coding from raw medical records. The system enables interactive EHR upload, visualization of intermediate agent reasoning steps, and presentation of final disease and procedure codes, promoting transparency and usability.

The backend is built on Django to manage requests and orchestrate multi-stage agent inference. An OCR module based on PaddleOCR (Cui et al., 2025) is integrated to parse PDF-formatted EHRs. Figure 2 shows a screenshot of the system interface.

Method	Disease (ICD-10)	Procedure (ICD-9-CM-3)
Vanilla	I25.102; I10.x00x027; E11.900; J39.203; Z01.100	88.5500; 36.0700x004; 00.4000 00.6600x004
Few-shot	I25.102; I67.400; E11.900; J39.203; Z01.100	88.5500; 36.0700x004
CoT	I25.102; I10.x00x027; E11.900; J39.200x016	36.0700x004
CoT-SC	I25.102; I10.x00x027; E11.900; J39.200x016; Z01.100	36.0700x004; 39.9000x016
SFT	E11.900; H91.900x004; I25.102; J39.203; I10.x00x028	00.6600x008; 00.6600x004; 88.5500; 00.4000
JointCoder	I25.102; I10.x00x028; E11.900; J39.200x016; H91.900x002	36.0700; 00.6600x004; 88.5500; 00.4500; 00.4100
Ground Truth	I25.102; I10.x00x028; E11.900; J39.200x016; H91.900x002	36.0700; 00.6600x004; 88.5500; 00.4500; 00.4100

Green: Correct predictions; Red: Incorrect predictions.

Table 4: Case study comparing disease and procedure ICD prediction results of different methods on a real-world Chinese EHR example.

Stage	Disease Coding (ICD-10)	Procedure Coding (ICD-9-CM-3)
Standardization	冠状动脉粥样硬化性心脏病；慢性心力衰竭；心功能Ⅲ级；高血压病3级（极高危）；孤立性肺结节	经皮冠状动脉球囊扩张成形术；药物洗脱冠状动脉支架置入
Candidate Mining	——	经皮冠状动脉球囊扩张成形术；药物洗脱冠状动脉支架置入；单根导管的冠状动脉造影术
Prioritization	慢性心力衰竭；冠状动脉粥样硬化性心脏病；心功能Ⅲ级；高血压病3级（极高危）；孤立性肺结节	药物洗脱冠状动脉支架置入；经皮冠状动脉球囊扩张成形术；单根导管的冠状动脉造影术
Code Also	——	药物洗脱冠状动脉支架置入；经皮冠状动脉球囊扩张成形术；单根导管的冠状动脉造影术；单根血管操作；两根血管操作；三根血管操作；四根或更多根血管操作；置入一根血管的支架；置入两根血管的支架；置入三根血管的支架；置入四根或更多根血管的支架
Validation	慢性心力衰竭；冠状动脉粥样硬化性心脏病；心功能Ⅱ级；高血压病3级（极高危）；肺诊断性影像异常	药物洗脱冠状动脉支架置入；经皮冠状动脉球囊扩张成形术；单根导管的冠状动脉造影术；两根血管操作；置入一根血管的支架
ICD Code Mapping	I50.908；I25.103；I50.903；I10.x00x032；R91.x00x003	36.0700；00.6600x004；88.5500；00.4100；00.4500

Table 5: An illustrative example of the JointCoder workflow, showing intermediate outputs for disease and procedure coding at each stage.

Figure 2: Screenshot of the web-based JointCoder system interface.