From Documents to Spans: Code-Centric Learning for LLM-based ICD Coding

Anonymous ACL submission

Abstract

ICD coding is a critical yet challenging task in healthcare. Recently, LLM-based methods demonstrate stronger generalization than discriminative methods in ICD coding. However, fine-tuning LLMs for ICD coding faces three major challenges. First, existing public ICD coding datasets provide limited coverage of the ICD code space, restricting a model's ability to generalize to unseen codes. Second, naive fine-tuning diminishes the interpretability of LLMs, as few public datasets contain explicit supporting evidence for assigned codes. Third, ICD coding typically involves long clinical documents, making fine-tuning LLMs computationally expensive. To address these issues, we propose Code-Centric Learning, a training framework that shifts supervision from full clinical documents to scalable, short evidence spans. The key idea of this framework is that span-level classification improves LLMs' ability to perform document-level ICD coding. Our proposed framework consists of a mixed training strategy and code-centric data expansion, which effectively improve coding performance on ICD codes out of domain while preserving interpretability. With only the open-source Llama-3.1-8B model, our method outperforms or matches strong discriminative baselines and GPT-4.1–based generative methods, demonstrating its effectiveness and potential for fully automated ICD coding. Code is available at https://anonymous.4open.science/r/CCL-ICD.

1 Introduction

ICD codes play a foundational role in medical insurance reimbursement and health data analysis. ICD coding, the process of assigning ICD codes to each patient encounter, is a critically important task in modern healthcare systems. However, ICD coding remains an extremely challenging task. Even expert human coders frequently make errors (Burns et al., 2012; Horsky et al., 2018; Gan et al., 2025).

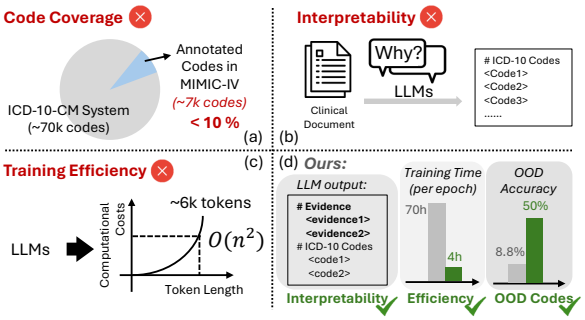

Figure 1: (a) Exsiting public datasets covers limited ICD codes; (b) Direct fine-tuning lacks interpretability; (c) Training cost of LLM on long documents is expensive; (d) Our solution hugely improves OOD code accuracy via efficient training, and preserves interpretabiliy.

First, multiple codes often apply to a single encounter, leading to pervasive undercoding. In addition, the large and fine-grained ICD vocabulary increases the likelihood of code confusion and misassignment. These challenges motivate extensive research on automated ICD coding.

Early ICD coding approaches adopt discriminative models with label attention mechanisms (Mullenbach et al., 2018; Huang et al., 2022; Edin et al., 2024). Recently, LLM-based methods gain attention due to their stronger generalization capabilities (Motzfeldt et al., 2025; Yuan et al., 2025). Among them, training-free approaches (Li et al., 2024; Motzfeldt et al., 2025) rely on carefully designed prompts or workflows, and place strong demands on the capability of the backbone LLM. However, such high-performance LLMs are often closed-source and require online access, making them incompatible with the strict data privacy and deployment constraints in clinical settings. In contrast, fine-tuning methods (Yuan et al., 2025) enable smaller, locally deployable LLMs to acquire ICD coding capabilities, making them more practical for real-world healthcare applications. Given clinical notes and a simple task prompt, the LLM is

optimized to output ICD codes along with their textual descriptions and outperforms discriminative models on out-of-distribution data.

However, this fine-tuning paradigm entails several issues: **limited code coverage**, **poor interpretability** and **low training efficiency** (shown in Figure 1). First, existing ICD coding datasets cover only a small subset of ICD codes. For example, even MIMIC-IV (Johnson et al., 2023) includes only about 10% of the 70K ICD-10-CM codes, which severely restricts generalization to unseen codes. Second, most ICD coding datasets provide ICD codes without supporting evidence; as a result, fine-tuning LLMs on such data encourages direct code prediction without explicit evidence grounding. This not only diminishes interpretability, but also precludes human review and correction of the underlying evidence, thereby limiting effective human–AI collaboration. Third, fine-tuning LLMs on long clinical documents is computationally expensive, considering the quadratic complexity with respect to input length. Collectively, these challenges significantly limit the applicability of fine-tuning LLMs for ICD coding.

To address the above issues, we propose a novel training framework, Code-Centric Learning (CCL). Unlike traditional paradigms that operate on entire clinical notes, our method centers on code-specific evidence spans. Intuitively, ICD coding can be decomposed into two sub-tasks: locating evidence and assigning codes. If an LLM is trained to assign the correct ICD code to an evidence span, it implicitly learns to recognize such evidence in a long clinical document. Consequently, **strengthening the evidence-level code assignment capability also enhances the model's ability to locate relevant evidence and assgin codes in a full note**.

Based on this intuition, CCL consists of two key strategies. First, we adopt a **mixed training** strategy that leverages a limited number of samples with annotated evidence spans together with a large collection of code-related evidence spans. Compared to direct fine-tuning on full clinical notes, this strategy provides explicit interpretability and reduces computational cost by focusing on short evidence spans instead of long clinical documents. Second, we propose a **code-centric data expansion** strategy. We extract code-specific evidence spans from public datasets based on annotated codes, and supplement them using official ICD coding resources. For codes unseen in both official knowledge bases and public datasets, we retrieve the closest codes and evidence to synthesize plausible evidence spans, ensuring full ICD code coverage. Our proposed training framework enables an 8B-scale LLM to outperform both discriminative models and strong GPT-4-based approaches on both in-domain and out-of-domain datasets, while additionally providing explicit interpretability and intervention capability.

Our main contributions are summarized as:

- We propose a fine-tuning framework for LLM-based ICD coding that simultaneously addresses three key limitations: (i) limited code coverage in public datasets, (ii) the lack of interpretability in standard fine-tuning, and (iii) the low training efficiency caused by long clinical documents.
- We introduce a novel training framework consisting of a mixed training strategy and a code-centric data expansion strategy, motivated by the central insight that **span-level classification yields transferable gains for document-level evidence extraction**.
- Our proposed framework enables an 8B-scale LLM to achieve competitive performance against both discriminative models and strong GPT-4-based approaches on both in-domain and out-of-domain datasets, while attaining state-of-the-art results on several key metrics. In addition, it provides explicit interpretability and supports human–AI collaboration.

2 Related Work

2.1 Discriminative methods

Label attention. Discriminative models have long dominated the ICD coding task, whose strong performance can be attributed to label attention mechanisms (Mullenbach et al., 2018), learning an independent query vector for each ICD code. The widely adopted state-of-the-art method is PLM-ICD (Huang et al., 2022) and its variant PLM-CA (Edin et al., 2024).

Knowledge injection. Building on label attention, several studies explore incorporating external ICD-related knowledge. DKEC (Ge et al., 2024) apply a graph network to encode knowledge from multiple sources. AKIL and MRR (Wang et al., 2024b,a) introduce auxiliary coding systems such as DRG, CPT, and procedure codes. Correlation (Luo et al., 2024) models relationships among ICD

codes. MSMN (Yuan et al., 2022) and MSAM (Gomes et al., 2024) ultilize synonyms to learn code representations. GKI-ICD (Zhang et al., 2025) injects code descriptions, synonyms, and hierarchical relations by synthesizing guidelines. ACE-ICD (Le et al., 2025) imrpoves medical terminology understanding via acronym expansion.

Interpretability. Traditional discriminative models operate as black-box systems, with limited interpretability. To address this challenge, MDACE (Cheng et al., 2023) re-annotated a subset of MIMIC-III (Johnson et al., 2016), in which expert coders performed multi-round cross-validation to relabel ICD codes and provided gold-standard evidence spans for each assignment. Building on MDACE, Edin et al. (2024) apply AttInGrad to map model predictions to evidence spans in the original text. AutoCodeDL (Wu et al., 2024) incorporates dictionary learning to decode dense embeddings into medical concepts. While discriminative models require additional mechanisms to produce evidence, generative models naturally support evidence-based coding.

2.2 Generative Methods

Training-free methods. Early works explored the use of off-the-shelf LLMs for ICD coding, designing prompts and workflows. Boyle et al. (2023) prompts the LLM to predict ICD codes in a hierarchical manner, from chapters and sections to specific codes. MAC (Li et al., 2024) prompts the LLM to perform as different roles, such as coder and physician, reducing coding errors via cross-role verification. MedCodER (Baksi et al., 2025) combines evidence extraction, candidate retrieval, and re-ranking, to overcome the large label space. Similarly, CLH (Motzfeldt et al., 2025) extracts evidence, retrieves candidate codes via the Alphabetic Index and Tabular List, and verifies them.

Fine-tuning methods. Recently, (Yuan et al., 2025) demonstrates that fine-tuning LLMs is more suitable for this task than training-free paradigms and proposes a verification module to fix mistakes. Nesterov et al. (2025) also validates this conclusion on their proposed Russian ICD coding benchmark. However, these methods typically perform supervised fine-tuning (SFT) directly on ICD coding datasets, which leads to low training efficiency on long documents, covers limited ICD code space, and lacks interpretability. Therefore, developing a more suitable fine-tuning paradigm for LLM-based ICD coding is particularly important.

3 Methodology

3.1 Overview

Fine-tuning LLMs for ICD coding is commonly performed at the document level. Given a clinical document x and its associated ICD code set C (followed by code description), conventional document-level training can be cast as supervised fine-tuning (SFT):

$$\min_{\theta} \ \mathcal{L}_{\text{SFT}}(f_\theta(x), C)\,, \tag{1}$$

where the LLM $f_\theta(\cdot)$ maps a long clinical document to a set of ICD codes under standard next-token prediction. However, this paradigm is inherently limited by three critical challenges: low training efficiency, high annotation noise, and poor interpretability (See Section 1).

To overcome these challenges, we propose a code-centric learning framework. We perform mixed SFT on two types of instances: (i) a full clinical document, annotated with evidence and ICD codes; (ii) an evidence span and a single ICD code, which can be formulated as:

$$\min_{\theta} \Big[\mathcal{L}_{\text{SFT}}\big(f_\theta(x), (E, C)\big) + \mathcal{L}_{\text{SFT}}\big(f_\theta(e), c\big)\Big], \tag{2}$$

where E denotes evidence spans supporting annotated ICD codes C, e denotes an evidence span with corresponding ICD code c. **The former forces the LLM to extract evidence before assigning codes, enhancing interpretability, while the latter consists of short evidence spans that are easy to acquire, which naturally address issues of limited code coverage and training efficiency.**

Defining (e, c) as an evidence-code pair,

$$(e, c) \sim \mathcal{D}, \tag{3}$$

where D is a multi-source knowledge base. To ensure full ICD code coverage, we construct $\mathcal{D}$ by integrating three tiers of evidence-code pairs: *gold*, *silver*, and *synthetic*, as:

$$\mathcal{D} = \mathcal{D}_{\text{gold}} \ \cup \ \mathcal{D}_{\text{silver}} \ \cup \ \mathcal{D}_{\text{syn}}, \tag{4}$$

in which $\mathcal{D}_{\text{gold}}$ consists of scarce, authoritative evidence-code pairs from official ICD resources. $\mathcal{D}_{\text{silver}}$ is mined from public datasets, yielding a larger collection of evidence-code pairs with broader coverage. For the remaining unseen codes, we construct $\mathcal{D}_{\text{syn}}$ by synthesizing evidence via

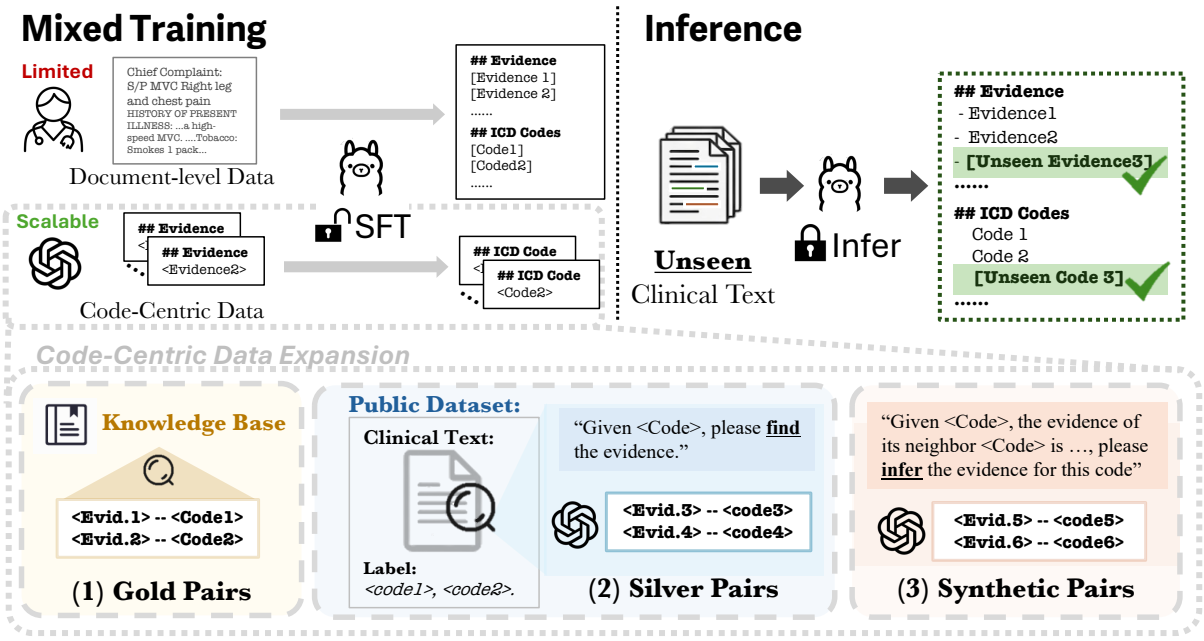

Figure 2: Overviw of code-centric learning framework. Mixed Training unifies document-level evidence-based ICD coding and span-level code knowledge injection within a single framework. Code-centric Data Expansion leverages large language models to extract and infer evidence spans for each code from diverse knowledge sources, addressing codes that lack evidence-annotated documents.

LLM, thereby completing coverage over the entire ICD code set. Below, we will describe the details of the mixed training strategy for Eq. 2, and the code-centric data expansion strategy for Eq. 4.

3.2 Mixed Training

Mixed training is designed to bridge the gap between scarce, high-quality document-level ICD coding data and large-scale but weakly supervised code-centric data. By jointly training on these two complementary supervision signals, the model learns to follow human-aligned evidence-based coding workflows while acquiring broad ICD knowledge that cannot be covered by limited expert annotations alone.

Document-level evidence-based ICD coding data refers to medical documents annotated with both ICD codes and supporting evidence. This type of data is very hard to obtain, and therefore extremely scarce and valuable. To our knowledge, MDACE (Cheng et al., 2023) is the only available public dataset that contains such kind of data.

For each clinical document, we first extract the human-annotated evidence spans, preserving their original order in the document. We then order the ICD codes accordingly, and augment them with their textual descriptions from the ICD-10 Tabular List. Finally, we convert text, evidence and codes into instruction-tuning samples using a unified prompt template (Appendix B). Note that evidence extraction and code assignment are performed jointly within a single generation process, rather than through staged or multi-step pipelines.

Despite MDACE's small scale, LLMs can learn robust evidence-based ICD coding behavior, as demonstrated in Section 5.2.

Code-centric learning data refers to evidence–code pairs, where the model takes an evidence span as input, instead of a full clinical document, and predicts the corresponding ICD code. The advantage of code-centric learning data is obvious: since each sample contains only a short evidence span and a single ICD code, it enables efficient training, and is scalable and substantially easier to denoise than document-level labels.

Such evidence–code pairs can be obtained from diverse sources. They may originate from high-quality, human-curated resources (e.g., official coding references) or be automatically extracted by LLMs from public ICD coding dataset. Section 3.3 describes how we systematically expand these pairs to increase code coverage.

Training and inference. We fine-tune the LLM using a mixture of document-level, evidence-based ICD coding data and large-scale code-centric learning data under a standard autoregressive super-

vised fine-tuning objective. The evidence-based data teaches the model to follow the human coding workflow and produce structured, interpretable outputs, while the code-centric data injects scalable code-specific knowledge and enables generalization beyond the limited code set covered by dense annotations.

$$\min_{\theta} \; \mathcal{L}_{\text{SFT}}(\theta) = \sum_{i=1}^{N} \sum_{t=1}^{T_i} -\log p_\theta\big(y_t^{(i)} \mid x^{(i)}, y_{<t}^{(i)}\big). \tag{5}$$

At inference time, we apply the same prompt schema used for evidence-based training (Appendix B). This unified formulation encourages the model to first identify relevant evidence spans and then assign ICD codes, while implicitly leveraging the code-level knowledge acquired during code-centric learning. As a result, the model produces ICD predictions that are both interpretable and accurate, without requiring evidence annotations at test time.

3.3 Code-centric Data Expansion

Under the code-centric learning paradigm, we organize training supervision around individual ICD codes rather than full clinical documents. Accordingly, we construct a multi-tier evidence–code knowledge base composed of gold pairs from human-curated ICD resources, silver pairs mined from noisy labeled datasets, and synthetic pairs inferred by LLMs to ensure full code coverage.

Gold pairs from human knowledge bases. In clinical practice, human coders routinely consult the Alphabetic Index and the Tabular List when assigning ICD codes (See Appendix A.3). These resources naturally form high-quality code-centric learning corpus, but had been overlooked by previous works. CLH (Motzfeldt et al., 2025) first incorporated these resources, treating them as an external corpus for retrieval-augmented generation. In our framework, we treat Alphabetic Index terms paired with their default ICD codes as gold evidence-code pairs.

Silver pairs from noisy labeled datasets. Most large-scale ICD datasets provide only document-level code labels without explicit evidence annotations. To align such data with evidence-based ICD coding, we construct silver evidence-code pairs via a two-stage LLM pipeline: document-level evidence extraction followed by code-level evidence consolidation.

In the first stage, given a clinical note and one of its assigned ICD codes, the LLM extracts a textual span that plausibly supports the code. We aggregate extracted spans across the dataset for each ICD code, retain unique evidence phrases, and record their frequencies.

$$\mathcal{E}_c = \{\, e \mid e = f_{\text{LLM}}(x, c),\; x \in \mathcal{X}_c \,\}, \tag{6}$$

where $\mathcal{X}_c$ denotes the set of clinical documents labeled with code c, $f_{\text{LLM}}(x, c)$ extracts supporting evidence spans from document x for code c, yielding a large evidence set $\mathcal{E}_c$. g_{LLM} then summarizes $\mathcal{E}_c$ into a small set of representative (typical) evidence expressions $\tilde{\mathcal{E}}_c$.

In the second stage, given an ICD code and its frequency-ranked evidence candidates, the LLM infers a small set of representative evidence expressions for that code, forming the silver dataset $\mathcal{D}_{\text{silver}}$.

$$\tilde{\mathcal{E}}_c = f_{\text{LLM}}\big(c,\; \mathcal{E}_c\big), \tag{7}$$

where f_{LLM} denotes the LLM, c denotes the target ICD code, and $\mathcal{E}$ denotes the evidence candidates.

Synthetic pairs inferred by LLMs. Despite combining gold and silver pairs, many ICD codes remain uncovered in public datasets. To achieve full code coverage, we synthesize evidence-code pairs using an LLM guided by ICD knowledge.

For each uncovered target ICD code, we retrieve its nearest neighbor code in the ICD-10-CM hierarchy and related information of this nearest code. Conditioned on this information, the LLM generates concise clinical evidence expressions that plausibly support the target code, forming the synthetic dataset $\mathcal{D}_{\text{syn}}$:

$$e = f_{\text{LLM}}\big(c,\; c^*,\; \mathcal{K}(c^*)\big), \tag{8}$$

where c denotes the target ICD code, c^* denotes its nearest ICD code, and $\mathcal{K}(c^*)$ represents the associated ICD knowledge of c^*, i.e. potential evidence from gold pairs and silver pairs. These synthetic pairs complement gold and silver data, resulting in a code-centric knowledge base with complete ICD coverage.

Finally, we mix gold, silver and synthetic evidence-code pairs, and convert them into the same prompt format as evidence-based ICD coding, to form a large, high-quality instruction-tuning dataset.

4 Experiments

4.1 Dataset

MIMIC-IV (Johnson et al., 2023) is currently the largest publicly available dataset annotated with ICD-10 codes. However, it contains only ICU clinical notes, while the ICD codes span the entire patient encounter. As a result, many codes are not supported by the available text (Cheng et al., 2023; Edin et al., 2023; Yuan et al., 2025), making MIMIC-IV unsuitable as a reliable benchmark. We therefore treat MIMIC as a large but noise-prone training dataset, and rely on high-quality external benchmarks to measure true ICD coding performance. Since some of the baselines are trained on its training set, our method uses the same training set to extract code-centric data, following Edin et al. (2024)'s split, to ensure fair comparison.

MDACE (Cheng et al., 2023) is an expert-annotated subset of MIMIC-III (Johnson et al., 2016), containing gold-standard evidence span annotations. We follow its official data split, using the training set for document-level evidence-based ICD coding, and using its test set for evaluation.

ACI-Bench (Yim et al., 2023) is a synthetic dataset of clinical notes, based on which Yuan et al. (2025) constructs a new double expert-annotated ICD-10-CM coding benchmark.

In summary, our training primarily relies on MIMIC-IV, supplemented with MDACE. Evaluation is conducted on MDACE and ACI-Bench, with the latter providing a more out-of-distribution benchmark to assess generalization.

4.2 Evaluation

For discriminative models, we use the validated-optimal threshold. For generative models, we extract the alphanumeric code component (letter followed by digits) from LLM's text output. We filter out codes not present in the test-set label space. Since generative models do not provide probability scores, AUC cannot be computed. We therefore report micro-F1, macro-F1, Precision, and Recall as our primary evaluation metrics.

4.3 Baselines

Discriminative Methods. PLM-ICD (Huang et al., 2022) is a BERT-based model combined with label attention. PLM-CA (Edin et al., 2024) is its variant, replacing label attention with cross attenion. GKI-ICD (Zhang et al., 2025) is a knowledge injected version of PLM-CA. We finetune these models on MIMIC-IV-ICD10 dataset, and find the optimal threshold.

Generative methods. For closed-source LLMs, we include Chain-of-Thought (CoT) (Wei et al., 2022) and its self-consistency variant (CoT-SC) (Wang et al., 2022), as well as task-specific methods, MAC (Li et al., 2024) and CLH (Motzfeldt et al., 2025). We additionally include CoT and SFT of the same backbone LLM. For SFT, we adopt the optimal configuration from Yuan et al. (2025), using code and description as output format.

5 Results

5.1 Comparison with SOTA models

Table 1 shows that our method (CCL) achieves strong performance using only Llama3.1-8B. **Compared with methods of similar scale, CCL substantially outperforms** prompt-based CoT and improves over standard SFT across both in-domain (MDACE) and out-of-domain (ACI-Bench) benchmarks. Despite its small model size, CCL achieves performance **comparable to GPT-4.1–based methods, and even surpasses** them on several key metrics (e.g., Micro-F1 on MDACE and competitive Micro-/Macro-F1 on ACI-Bench), while remaining competitive on the others.

Importantly, CCL also provides explicit evidence for its predicted ICD codes, enabling human coders to review and revise the evidence, and our experiments in 2 show that such revisions lead to measurable performance improvements. Overall, CCL strikes an effective balance between performance, efficiency, and interpretability.

5.2 Ablation Study

We conduct ablation studies on the MDACE dataset using Llama-3.1-8B as the backbone model.

Evidence-based fine-tuing is more effective than code-only fine-tuning. First, even with a small amount of fine-tuning data, i.e. MDACE training set, LLMs achieve clear performance gains over CoT without fine-tuning, demonstrating that fine-tuning is necessary and effective for ICD coding. Second, on the same amount of training data, incorporating human-annotated evidence spans into ground truth (i.e. train LLMs to extract evidence before assgining codes) leads to larger improvements than code-only fine-tuning. This behavior contrasts with prior findings on discriminative models, where evidence-span supervision does not necessarily improve coding accuracy (Cheng et al., 2023).

Method	Backbone	MDACE (In Domain)				ACI-Bench (Out of Domain)			
		Micro-F1	Macro-F1	Recall	Precision	Micro-F1	Macro-F1	Recall	Precision
PLM-ICD (Huang et al., 2022)	RoBERTa (120M)	50.6	26.5	**66.6**	40.8	39.3	18.3	58.1	39.6
PLM-CA (Edin et al., 2024)	RoBERTa (120M)	50.0	25.8	65.3	40.4	25.1	11.1	64.8	15.5
GKI-ICD (Zhang et al., 2025)	RoBERTa (120M)	50.4	26.3	64.3	41.0	47.8	25.8	62.0	39.0
CoT (Wei et al., 2022)	GPT-4.1	57.0	35.8	59.7	54.4	61.9	**48.1**	67.6	57.0
CoT-SC (Wang et al., 2022)	GPT-4.1	57.1	36.2	56.1	58.2	61.4	47.4	67.6	56.2
MAC (Li et al., 2024)	GPT-4.1	48.1	31.0	61.2	39.7	51.5	45.6	**69.6**	40.9
CLH (Motzfeldt et al., 2025)	Qwen3-235BA22B	56.3	**43.7**	55.2	57.6	**67.1**	47.7	66.1	68.1
CoT (Wei et al., 2022)	Llama3.1-8B	26.5	9.1	20.2	38.9	43.4	12.4	36.4	53.6
SFT (Yuan et al., 2025)	Llama3.1-8B	57.4	34.5	60.1	55.0	63.4	38.9	57.9	**70.1**
CCL (Ours)	Llama3.1-8B	**59.3**	35.2	56.4	**62.5**	65.8	47.1	64.8	67.0

Table 1: Comparison with previous SOTA methods on in domain and out of domain benchamrks.

Methods	Micro-F1	Macro-F1	Recall	Precision
w/o fine-tuning	26.5	9.1	20.2	38.9
Code-only	42.3	17.0	37.8	48.2
Evidence & Code	49.7	21.4	47.0	52.8
CCL (Gold-only)	53.4	25.2	57.2	50.1
CCL (Ours)	59.3	35.2	56.4	62.5
w/ human evidence	78.0	54.8	73.5	83.0

Table 2: Ablation Study on MDACE Dataset. Code-only and Code & Evidence use MDACE training set, whereas CCL additionally incorporates evidence spans. w/ Human Evidence indicates that, during testing, human-annotated evidence is added to the input, allowing the LLM to continue assigning codes.

Methods	Micro-F1	Macro-F1	Recall	Precision
Llama3.1-8B	59.3	35.2	56.4	62.5
Llama-3.2-1B	40.8	16.0	35.2	48.6
Llama-3.2-3B	53.2	26.0	47.4	60.6
Qwen2.5-0.5B	31.8	10.6	24.9	44.2
Qwen2.5-1.5B	50.6	24.1	45.0	57.8
Qwen2.5-7B	54.2	27.4	48.6	61.2

Table 3: Performance of different LLM under CCL Framework on MDACE Dataset.

Code-centric learning effectively improves coding performance. We first conduct a simplified experiment, using only gold pairs from Alphabetic Index. As shown in Table 2, CCL on gold pairs already yields a clear improvement in coding performance. Furthermore, we extract or synthesize code-specific evidence spans, i.e. silver and synthetic pairs, to achieve full code coverage, which leads to further gains. These results highlight the effectiveness of code-centric learning and code-centric data expansion.

Human-AI Collaboration. Our paradigm explicitly extracts evidence before code assignment, enabling human-in-the-loop ICD coding by allowing clinicians to review and revise LLM-generated evidence. As shown in Table 5.2, replacing model-generated evidence with human-annotated evidence yields substantial performance improvements, highlighting the interpretability, controllability, and practical applicability of our approach.

5.3 Generality and Scalability

We conduct extensive experiments across LLMs with different architectures and parameter scales. Due to hardware constraints, we are limited to fine-tuning models with up to 8B parameters.

Generality across architectures. As shown in Table 3, the proposed framework performs consistently across both Llama and Qwen backbones, demonstrating robust cross-architecture generalization. Across comparable model scales, Llama models consistently outperform Qwen models, which may be due to Llama being primarily pretrained on English data.

Scalability with model size. Within each model family, performance improves steadily as model size increases. Both Llama-3.2 and Qwen2.5 models exhibit clear gains in Micro-F1 and Macro-F1 when scaling from smaller to larger variants. The consistent scaling behavior across architectures further supports the scalability of the approach.

5.4 Analysis of Code-Centric Learning

Code-centric Learning enables the model to acquire knowledge of codes beyond the coverage of the document-level data. As illustrated in Figure 3 (a), we categorize the codes in the test set into two groups: (1) codes covered by the document-level data, for which the model explicitly learns to extract supporting evidence and assign ICD codes, and (2) codes learned exclusively through code-centric learning, for which the model has only been trained to assign codes without explicit evidence extraction. The results in Figure 3 (c) demonstrate that, under the mixed training strategy, the code

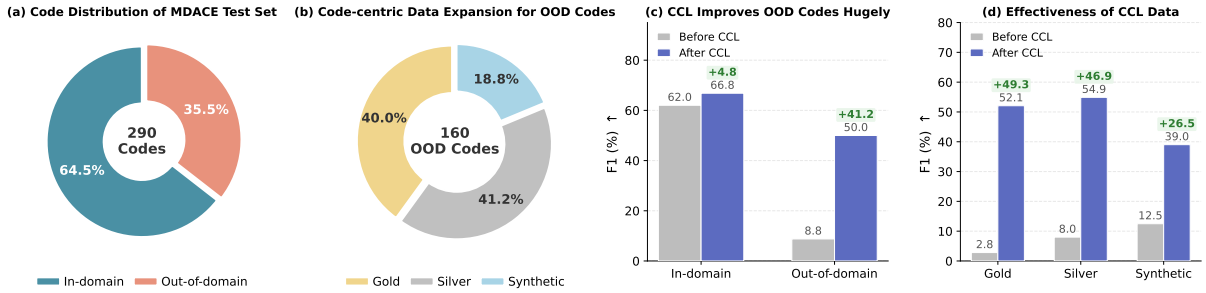

Figure 3: (a) Test-set codes are partitioned into in-domain and out-of-domain (unseen in the document-level training set). (b) Sources of spans constructed for OOD codes, with proportions of three strategies. (c) Our method improves OOD performance using spans only, without documents. (d) Each strategy contributes to improved coding accuracy.

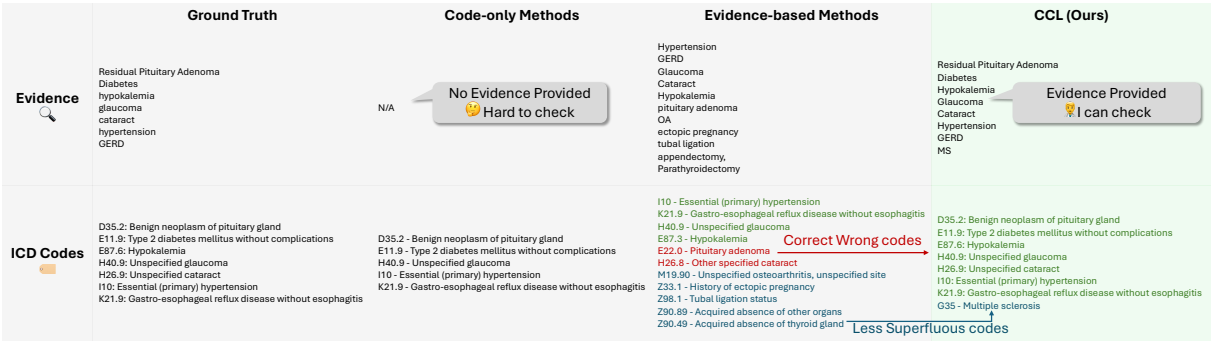

Figure 4: An example from the test set. Code-only methods cannot generate evidence, making the results difficult for humans to evaluate and revise. Evidence-based methods suffer from limited evidence-annotated data for fine-tuing, and therefore achieve lower accuracy. Our method balances interpretability and accuracy, producing evidence and ICD codes that are highly consistent with human annotations.

knowledge learned during code-centric learning can be transferred to document-level ICD coding, which demonstrates the effectiveness of our proposed hybrid training strategy.

Code-Centric Data Expansion effectively broadens the code coverage of document-level ICD coding. Overlall, CCL data derived from all three knowledge sources consistently improve OOD coding performance, demonstrating that document-level ICD coding can be effectively enhanced using span-level supervision alone. Further analysis across knowledge sources shows that evidence–code pairs mined from official guidelines and public datasets yield larger performance gains, while synthetic data also contributes to OOD code recognition, albeit to a relatively smaller extent.

5.5 Case Study

Figure 4 shows an example from the test set. Unlike **code-only methods** that achieve high accuracy but only output final codes without supporting evidence, and **evidence-based methods** whose accuracy is constrained by limited evidence-annotated data, our approach is able to generate interpretable evidence while maintaining accurate ICD coding. In this case, the predicted ICD codes and supporting evidence are highly consistent with the ground truth, which demonstrates that our method effectively **balances interpretability and accuracy**.

6 Conclusion

In this paper, we identify a key finding: span-level learning can effectively boost document-level ICD coding. Based on this observation, we propose a new LLM fine-tuning paradigm for ICD coding, termed code-centric learning. This training paradigm incorporates mixed training and code-centric data expansion, and simultaneously addresses three long-standing challenges in this domain: limited code coverage, lack of interpretability and inefficient training on long documents. Our paradigm is efficient, scalable and better suited for real-world deployment in collaboration with human coders. We hope that the community will adopt this new paradigm and advance the long-standing goal of fully automated ICD coding.

Limitations

Data quality. The scarcity of high-quality ICD coding benchmarks remains a fundamental challenge for the research community. Although MIMIC-III and MIMIC-IV are widely used, many studies have repeatedly highlighted their annotation inconsistencies and data quality issues (Cheng et al., 2023; Edin et al., 2023; Yuan et al., 2025). Even comparatively higher-quality benchmarks such as MDACE still contain labeling errors (Khadka et al., 2025). We therefore call for the development of more reliable, rigorously curated ICD coding benchmarks.

Simiplified setting. First, existing datasets cover only a limited subset of the ICD ontology, typically a few thousand codes, which fall far short of the tens of thousands of codes present in the full ICD-10 system (Motzfeldt et al., 2025). Furthermore, practical ICD coding is substantially more complex than the simplified one-to-one mapping between evidence spans and codes. ICD codes exhibit hierarchical, inclusive, and mutually exclusive relationships, and these interactions strongly influence how evidence should be identified and grouped (CMS and NCHS, 2025). Modeling such relational structures poses significant technical challenges. Currently, the community still lacks such comprehensive benchmarks. Addressing these challenges requires collective effort of the community.

Ethics Statement

We use the publicly available clinical dataset. We do not see any ethics issues.

References

Krishanu Das Baksi, Elijah Soba, John J Higgins, Ravi Saini, Jaden Wood, Jane Cook, Jack I Scott, Nirmala Pudota, Tim Weninger, Edward Bowen, and Sanmitra Bhattacharya. 2025. MedCodER: A generative AI assistant for medical coding. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pages 449–459, Albuquerque, New Mexico. Association for Computational Linguistics.

Joseph Boyle, Antanas Kascenas, Pat Lok, Maria Liakata, and Alison O'Neil. 2023. Automated clinical coding using off-the-shelf large language models. In *Deep Generative Models for Health Workshop NeurIPS 2023*.

Elaine M Burns, E Rigby, R Mamidanna, A Bottle, P Aylin, P Ziprin, and OD Faiz. 2012. Systematic review of discharge coding accuracy. *Journal of public health*, 34(1):138–148.

Hua Cheng, Rana Jafari, April Russell, Russell Klopfer, Edmond Lu, Benjamin Striner, and Matthew R Gormley. 2023. Mdace: Mimic documents annotated with code evidence. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7534–7550.

CMS and NCHS. 2025. ICD-10-CM Official Guidelines for Coding and Reporting (FY 2025). Technical Report 10, The Centers for Medicare and Medicaid Services (CMS) and the National Center for Health Statistics (NCHS).

Joakim Edin, Alexander Junge, Jakob D Havtorn, Lasse Borgholt, Maria Maistro, Tuukka Ruotsalo, and Lars Maaløe. 2023. Automated medical coding on mimic-iii and mimic-iv: a critical review and replicability study. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2572–2582.

Joakim Edin, Maria Maistro, Lars Maaløe, Lasse Borgholt, Jakob Drachmann Havtorn, and Tuukka Ruotsalo. 2024. An unsupervised approach to achieve supervised-level explainability in healthcare records. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4869–4890, Miami, Florida, USA. Association for Computational Linguistics.

Yidong Gan, Maciej Rybinski, Ben Hachey, and Jonathan K. Kummerfeld. 2025. Aligning AI research with the needs of clinical coding workflows: Eight recommendations based on US data analysis and critical review. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 909–922, Vienna, Austria. Association for Computational Linguistics.

Xueren Ge, Abhishek Satpathy, Ronald Dean Williams, John Stankovic, and Homa Alemzadeh. 2024. DKEC: Domain knowledge enhanced multi-label classification for diagnosis prediction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12798–12813, Miami, Florida, USA. Association for Computational Linguistics.

Goncalo Gomes, Isabel Coutinho, and Bruno Martins. 2024. Accurate and well-calibrated ICD code assignment through attention over diverse label embeddings. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2302–2315, St. Julian's, Malta. Association for Computational Linguistics.

Jan Horsky, Elizabeth A Drucker, and Harley Z Ramelson. 2018. Accuracy and completeness of clinical coding using icd-10 for ambulatory visits. In *AMIA annual symposium proceedings*, volume 2017, page 912.

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, Seattle, WA. Association for Computational Linguistics.

Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. 2023. Mimic-iv, a freely accessible electronic health record dataset. *Scientific data*, 10(1):1.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

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

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Tuan-Dung Le, Shohreh Haddadan, and Thanh Q. Thieu. 2025. Ace-icd: Acronym expansion as data augmentation for automated icd coding. *Preprint*, arXiv:2511.07311.

Rumeng Li, Xun Wang, and Hong Yu. 2024. Exploring llm multi-agents for icd coding. *arXiv preprint arXiv:2406.15363*.

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 *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3997–4007, Torino, Italia. ELRA and ICCL.

Andreas Geert 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 *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 22612–22627, Suzhou, China. Association for Computational Linguistics.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1101–1111, New Orleans, Louisiana. Association for Computational Linguistics.

Alexandr Nesterov, Andrey Sakhovskiy, Ivan Sviridov, Airat Valiev, Vladimir Makharev, Petr Anokhin, Galina Zubkova, and Elena Tutubalina. 2025. RuCCoD: Towards automated ICD coding in Russian. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2558–2585, Suzhou, China. Association for Computational Linguistics.

Xindi Wang, Robert Mercer, and Frank Rudzicz. 2024a. Multi-stage retrieve and re-rank model for automatic medical coding recommendation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4881–4891, Mexico City, Mexico. Association for Computational Linguistics.

Xindi Wang, Robert E. Mercer, and Frank Rudzicz. 2024b. Auxiliary knowledge-induced learning for automatic multi-label medical document classification. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2006–2016, Torino, Italia. ELRA and ICCL.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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

Wen-wai Yim, Yujuan Fu, Asma Ben Abacha, Neal Snider, Thomas Lin, and Meliha Yetisgen. 2023. Aci-bench: a novel ambient clinical intelligence dataset for benchmarking automatic visit note generation. *Scientific data*, 10(1):586.

Moy Yuan, Han-Chin Shing, Mitch Strong, and Chaitanya Shivade. 2025. Toward reliable clinical coding with language models: Verification and lightweight adaptation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 173–184, Suzhou (China). Association for Computational Linguistics.

Zheng Yuan, Chuanqi Tan, and Songfang Huang. 2022. Code synonyms do matter: Multiple synonyms matching network for automatic ICD coding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2:*

Short Papers), pages 808–814, Dublin, Ireland. Association for Computational Linguistics.

Xu Zhang, Kun Zhang, Wenxin Ma, Rongsheng Wang, Chenxu Wu, Yingtai Li, and S Kevin Zhou. 2025. A general knowledge injection framework for ICD coding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7180–7189, Vienna, Austria. Association for Computational Linguistics.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Appendix

A ICD Coding Background

A.1 ICD coding task.

The ICD Coding task recognizes diseases, symptoms, conditions and procedures in a medical document, including discharge summaries, progress notes, and operative reports, and assign standardized ICD (International Classification of Diseases) codes to them. This task plays a critical role in healthcare administration, clinical statistics, reimbursement systems, and medical research.

From a computational perspective, ICD Coding is commonly formulated as a text-to-code prediction problem. Given a patient-level clinical document, the model is required to output a set of ICD codes. The task is characterized by a large label space, hierarchical code structures, and severe label imbalance, which together make ICD Coding a challenging and distinctive problem in clinical natural language processing.

A.2 Distinction from Related Tasks

Although ICD Coding seams to share similarities with several well-studied tasks, its objectives and constraints differ substantially from those tasks, which may lead to confusion.

Difference from Diagnosis. Diagnosis aims to infer or determine what diseases a patient has, often involving clinical reasoning, uncertainty management, and causal inference. In contrast, ICD Coding does not seek to generate new diagnostic conclusions. Instead, it focuses on assigning standardized codes based solely on diagnoses and clinical facts that have already been documented by healthcare professionals. Therefore, **ICD coding should be viewed as an information standardization task rather than a diagnostic or decision-making task**.

Difference from Multi-label Text Classification. In conventional multi-label settings, labels are typically assumed to be conditionally independent given the input text, and prediction is treated as a parallel, order-agnostic decision process. In contrast, ICD Coding involves strong dependencies among codes. In real-world coding practice, human coders do not assign all codes independently or simultaneously. Instead, they follow official coding guidelines to sequentially identify the principal diagnosis, secondary diagnoses, supplementary conditions, and procedures, with each step constraining and informing subsequent coding decisions. This process is inherently procedural and generative, rather than purely discriminative.

Difference from Information Extraction. Information extraction tasks typically focus on identifying entities, relations, or events explicitly mentioned in text. By contrast, ICD Coding requires the prediction of standardized code identifiers rather than text spans, and often relies on a holistic understanding of the entire document rather than localized entity mentions.

A.3 Authoritative Resources in ICD Coding

Alphabetic Index. The Alphabetic Index maps various synonyms, abbreviations, and lexical variants to candidate ICD codes, thereby bridging the gap between natural language expressions and standardized code identifiers. Importantly, the codes suggested by the Alphabetic Index are not definitive; rather, they represent preliminary references that must be further validated.

Tabular List. The Tabular List is the authoritative, structured listing of all valid ICD codes, organized by chapters, categories, subcategories, and extensions. Each code entry in the Tabular List is accompanied by a formal definition and may include additional annotations such as inclusion terms, exclusion notes, code-first instructions, and combination code indicators. Coders are required to confirm all codes suggested by the Alphabetic Index against the Tabular List before assignment.

Coding Guidelines. The Coding Guidelines provide a comprehensive set of rules and conventions that govern how ICD codes should be applied in practice. Guidelines often specify conditional logic (e.g., "code first," "use additional code," or "do not code separately") and clarify how multiple diagnoses or clinical conditions should be represented in a single episode.

In practical ICD coding workflows, these resources are used in a complementary and sequential manner. The Alphabetic Index supports initial term-to-code lookup, the Tabular List determines valid and precise code selection, and the Coding Guidelines regulate how codes are combined, ordered, and reported.

B Prompts

B.1 Prompts for Code-centric Learning Framework

In this section, we present all the prompts used in our approach. Specifically, we describe the prompts for Mixed Training and Code-centric Data Expansion.

Mixed Training relies on two types of data formats: (1) document-level evidence-based ICD coding data, and (2) span-level data designed for code-centric learning. We show the prompts of these two different tasks in Table 4.

For Code-centric Data Expansion, we show the prompts used to construct Silver Pairs and Synthetic Pairs, as Gold Pairs are primarily obtained from the Official Alphabetic Index.

To construct Silver Pairs, we employ LLaMA 3.1-70B to mine all supporting evidence from each MIMIC-IV sample, followed by deduplication and refinement of the evidence associated with each ICD code. We show the used prompts in Table 6.

For Synthetic Pairs, we use GPT-5.1 to infer unseen ICD codes based on existing Gold and Silver Pairs. We show the prompts in Table 5.

B.2 Prompts for Baselines

In this section, we present all the prompts used by the generative baselines, as shown in Table 7.

For Chain-of-Thought (CoT), we adopt the standard CoT prompting strategy.

For CoT-SC, we use the same prompt as CoT, but retain only those ICD codes that appear in at least three out of five reasoning runs.

For MAC, we make minor modifications to the original prompt to adapt it from ICD-9 to ICD-10, as the original method was evaluated primarily on ICD-9.

For CLH, we use the official open-source implementation and apply it directly to our dataset.

For the SFT on MIMICIV and Code-only ICD Coding setting on MDACE in the ablation study, we use the same prompt, following Yuan et al. (2025) to add descriptions after ICD codes.

C Implementation Details

For LLM inference, we use vLLM (Kwon et al., 2023). For SFT, we use LLaMa Factory (Zheng et al., 2024). All the experiments are implementated on a single H20 GPU with 96GB of VRAM.

Table 4: Prompt templates used for Mixed Training

Data Type	**Prompt Template**
Document-level ICD coding Data	Task: You are a clinical coding assistant. Your task is to analyze the provided clinical note, first extract all relevant clinical evidences that supports diagnostic coding, and then output the corresponding ICD−10−CM codes. Example ### Clinical Note: ... ### Evidence CAD COPD Anemia ### ICD−10−CM Codes I25.10 − Atherosclerotic heart disease of native coronary artery without angina pectoris J44.9 − Chronic obstructive pulmonary disease, unspecified D62 − Acute posthemorrhagic anemia −−− ### Clinical Note: \{text\}
Span-level Code-Centric Learning Data	### Evidence: \{evidence\} ### ICD−10−CM Codes:

Table 5: Prompt templates used for Code-Centric Data Expansion (Silver Pairs)

Task	**Prompt Template**
Evidence Extraion	You are a professional ICD−10−CM coder. Your task is to extract the ∗verbatim minimal text spans∗ that supports each ICD−10−CM code. If no explicit evidence exists in the note, output: "No evidence found". −−− Example ### Clinical Note: ... ### ICD−10−CM Codes ... ### Evidence I25.10 − Atherosclerotic heart disease of native coronary artery without angina pectoris > CAD J44.9 − Chronic obstructive pulmonary disease, unspecified > COPD D62 − Acute posthemorrhagic anemia > Anemia −−− ### Clinical Note {text} ### ICD−10−CM Codes {diagnosis_codes} ### Evidence
Evidence Refinement	You are a professional ICD−10−CM coder. Your task is to update and refine the Evidence Set for the ICD−10−CM code below. Follow these rules: 1. Only keep the ∗∗most essential∗∗ evidence that clearly supports this code. 2. You may reference the Alphabetic Index terms, but you do not need to match them exactly. 3. Use the ∗∗Original Evidence Set∗∗ as the base. − If the MIMIC−IV evidence contains new, meaningful, or more specific expressions, add them. − If not, keep the existing evidence unchanged. 4. Remove duplicates and unify phrasing into ∗∗clear, concise, canonical∗∗ clinical expressions. 5. Output the ∗∗updated Evidence Set only∗∗, as a bullet list. No explanation. −−− ### ICD−10−CM Code {code} ### Alphabetic Index Term {alphabetic_index_term} ### Original Evidence Set {evidence_set} ### New Evidence from MIMIC−IV {mimiciv_evidence} ### Updated Evidence Set −

Table 6: Prompt templates used for Code-Centric Data Expansion (Synthetic Pairs)

Task	**Prompt Template**
Synthesize Evidence	You are a professional ICD−10−CM coding and clinical documentation expert. Your task is to synthesize a focused, audit−defensible list of clinical evidence terms that directly support assignment of the ICD−10−CM code: {code}. Definition of evidence: Evidence refers only to clinical findings or documentation elements that materially support the diagnosis represented by the code. Available references: {reference} Instructions: − Use the parent and sibling codes to understand diagnostic scope. − Infer conservatively based on ICD−10−CM conventions and real−world clinical documentation patterns. − Prioritize diagnostic−confirmatory evidence (e.g., imaging findings, explicit diagnoses, anatomical localization). Do NOT include: − Mechanism of injury or accident descriptions − General symptoms or nonspecific complaints − Treatment, procedures, immobilization, or care plans − Encounter setting or workflow details − Redundant negative statements unless required to distinguish code type Unspecified code rule: − If the code is unspecified, do NOT introduce inferred specificity (e.g., displacement, fracture pattern, severity). Output constraints: − Consolidate overlapping or synonymous terms. − Stop generating new items once additional terms no longer add distinct coding value. Output format: − <evidence term> − <evidence term> ...

Table 7: Prompt templates used for Generative Baselines

Baselines	**Prompt Template**
CoT	You are a clinical coding assistant. Your task is to analyze the provided clinical note, and then output the corresponding ICD−10−CM codes. ### Clinical Note: {text} Let's think step by step.
MAC-coder	You are an ICD−10 coder. You assign ICD−10 codes to the discharge summary based on the clinical care that the patients received. You cite the discharge summary as evidence when needed. You assign as manyas possible ICD−10 codes and explain the reasons for each code. The discharge summary is: {text}
MAC-reviewer	You are a reviewer. You will check the ICD−10 codes assigned by the coder. You can use the ICD−10 dictionary for guidance. Your role is to ensure that the assigned ICD−10 codes are correct. You assign all possible ICD−10 codes and explain the reasons for each code. The discharge summary is: {text} The ICD−10 codes assigned by the coder are: {coder_pred}
MAC-physician	You are a physician who treats patients. You strive to provide the best service to each patient. You document your findings, interventions and results in the discharge summary note. You check all assigned ICD−10 codes and explain the reasons for each code. The discharge summary is: {text} The ICD−10 codes assigned by the coder are: {reviewer_pred}
MAC-patient	You are a patient who receieved treatment at the hospital. You cooperate fully with thehealth care system to receive the best service possible. You also check the ICD−10 codes to avoid being overbilled. You check all assigned ICD−10 codes and explain the reasons for each code. The discharge summary is: {text} The ICD−10 codes assigned by the coder are: {reviewer_pred}
MAC-adjustor	When a patient or a physician has different thoughts about the ICD−10 codes, you will review the discharge summary and the ICD codes assigned by the coder and checked by the reviewer. You can add, remove the assigned codes to make them accurate. You can consult the ICD−10 dictionary for assistance. Your duty is to ensure that the assigned ICD−10 codes are valid and exact. You assign all possible ICD−10codes and explain the reasons for each code. The discharge summary is {text} The ICD−10 codes assigned by the physician are {physician_pred} The ICD−10 codes assigned by the patient are {patient_pred} The ICD−10 codes assigned by the coder are {coder_pred} The ICD−10 codes checked by the reviewer are {reviewer_pred}
SFT / Code-only	You are a clinical coding assistant. Your task is to analyze the provided clinical note, and then output the corresponding ICD−10−CM codes. ### Clinical Note: {text}