Code Like Humans: A Multi-Agent Solution for Medical Coding

Andreas Motzfeldt[1,2] **Joakim Edin**[1,3] **Casper L. Christensen**[1]
Christian Hardmeier[2] **Lars Maaløe**[1] **Anna Rogers**[2]

[1]Corti.ai [2]IT University of Denmark [3]University of Copenhagen

Abstract

In medical coding, experts map unstructured clinical notes to alphanumeric codes for diagnoses and procedures. We introduce *Code Like Humans*: a new agentic framework for medical coding with large language models. It implements official coding guidelines for human experts, and it is the first solution that can support the full ICD-10 coding system (+70K labels). It achieves the best performance to date on rare diagnosis codes. Fine-tuned discriminative classifiers retain an advantage for high-frequency codes, to which they are limited. Towards future work, we also contribute an analysis of system performance and identify its 'blind spots' (codes that are systematically undercoded).

Code: https://github.com/MotzWanted/codeseeker

1 Introduction

For statistical and billing purposes, unstructured clinical notes need to be mapped to medical codes: alphanumeric codes of diagnoses or procedures (Chandawarkar et al., 2024). The International Classification of Diseases (ICD) is the most widely used system for diagnosis codes. Unfortunately, it is a time-intensive task,[1] in which errors can cause patient mistreatment and lost revenue (Gao et al., 2024; Gaffney et al., 2022).

Progress in NLP methods for medical coding has been stagnant in recent years. There have been several attempts to apply large language models (LLMs) (Boyle et al., 2023; Falis et al., 2024), but they have yet to yield improvements over the 2022 state-of-the-art (Huang et al., 2022) that used the BERT architecture (Devlin et al., 2019).

We hypothesize that one reason for the lack of progress may be the disconnect between current modeling approaches and how human medical coders work (Gan et al., 2025). Human coders begin not from memory but from the *alphabetical index* of ICD, a list that exhaustively maps clinical terms to codes capturing synonyms, eponyms, and contextual cues. Although this resource is an authoritative source of information used by medical coders to narrow down the search space, the current approaches do not consider it in any way.

To address this, we propose *CLH* (Code Like Humans): an LLM-based agentic framework[2] that leverages ICD resources such as the alphabetic index. *CLH* is inspired by the medical coding workflow developed for healthcare professionals in the UK (NHS England, 2023; CMS and NCHS, 2025). Our approach is summarized in Figure 1.

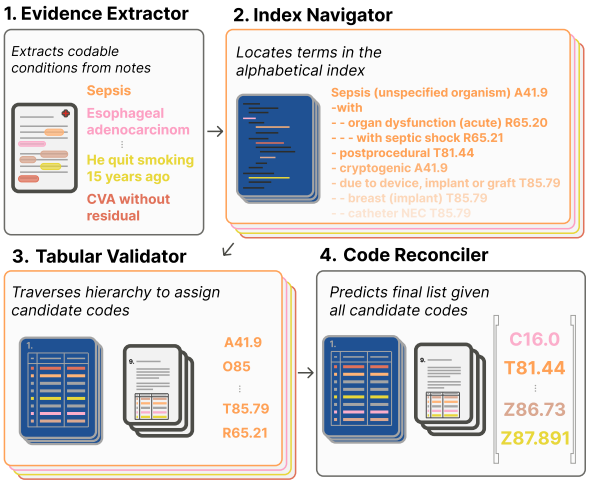

Figure 1: Overview of *Code-Like-Humans*, our agentic framework whose structure mirrors the *Analyze-Locate-Assign-Verify* approach of the UK National Health Service. The four agents sequentially emulate how medical coders extract evidence, navigate the alphabetical index, validate the ICD hierarchy, and reconcile coding conventions to 'translate' clinical notes into ICD codes.

[1]It takes a professional inpatient human coder approximately 30 minutes per case on average (Chen et al., 2021).

[2]An LLM-based agentic system expands the functionality of a stand-alone LLM in a goal-oriented manner to plan and take actions over time, interacting with tools, data sources, or other agents (Qiu et al., 2024).

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 22612–22627
November 4-9, 2025 ©2025 Association for Computational Linguistics

Our contributions are:

1. We propose *CLH*, an LLM-based agentic framework designed to mirror human medical coders by leveraging the same external resources: the alphabetic index, the ICD hierarchy, and the ICD guidelines.

2. We develop and publicly release an agentic implementation of *CLH* that can support the complete US ICD-10 system (70K labels). This solution achieves comparable performance with the state-of-the-art fine-tuned classifiers on rare codes, although the frequent codes still pose a challenge.

3. We provide an extensive analysis of our solution, identifying its difficulties in several areas of medical taxonomy for future work.

We conclude by discussing the overall readiness of the current solutions for medical coding. While in NLP this task has historically been approached from the *automation* perspective (i.e. end-to-end classification), at this stage, none of the existing approaches are ready for real-world deployment as a replacement for human coders. We argue that the task should generally be reconsidered as *assisting* the human coders, based on the identification of the pain points in their process that LLM technology can realistically help with.

2 Related Work and Novelty

Current approaches rely on encoder-decoder architectures that embed clinical notes and codes into separate embedding spaces. Label-wise attention with fine-tuning is typically used to align them (Mullenbach et al., 2018; Li and Yu, 2020). Prior work has sought to improve alignment by initializing embeddings with textual descriptions (Dong et al., 2020), synonyms (Yuan et al., 2022), or code co-occurrence (Xie et al., 2019; Cao et al., 2020). However, state-of-the-art results have been achieved using discriminative models, pre-trained on biomedical texts and then fine-tuned for classification (PLM-ICD) (Huang et al., 2022; Edin et al., 2023). This is the main baseline considered in this work.

These methods face three main challenges, listed below, together with our proposed solutions.

(i) Extremely large label set. The full ICD-10 coding system has over 70K possible labels to assign. In most of the prior work (e.g. Mullenbach et al., 2018; Li and Yu, 2020; Vu et al., 2020; Cao et al., 2020; Yuan et al., 2022), label-wise attention is filtered to codes present in MIMIC, 10% of the 70K ICD-10-CM codes. There are currently no public datasets that could verify the effectiveness of any approach for the full ICD-10 space. Furthermore, many prior studies report exclusively the performance on the top 50 codes (Gan et al., 2025). This setup simplifies the actual problem of medical coding, but it does not accurately reflect the needs of a real-world application.

Our solution. CLH retrieves codes from the ICD alphabetic index, enabling true open-set coding at inference. This makes *CLH* the first approach to cover[3] the full ICD-10 solution space.

(ii) Predicting codes with few or no training examples. When discriminative models are fine-tuned with standard cross-entropy on imbalanced data, random mini-batches mirror the label frequencies in the training set. The learned decision boundaries are biased toward frequent codes, under-predicting the rare codes. This bias might also lead to poor performance when code distributions differ across clinical settings (unfortunately, there are no public datasets that could be used to estimate the scope of this problem). Thus, such models only predict codes observed during training. Extending label-wise attention to the full solution space would result in negligible scores for unseen codes due to untrained weights, which is why in practice, the label set is a subset (see (i)). Recent works explored LLMs without fine-tuning for code extraction (Yang et al., 2023; Boyle et al., 2023; Gero et al., 2023), data augmentation for rare codes (Falis et al., 2024), and few-shot generative coding with gains on a constructed few-shot split of MIMIC-III (Yang et al., 2023), but none surpassed PLM-ICD.

Our solution. Rather than learning a label-frequency prior from training data, *CLH* uses a retrieval-induced distributional prior from the Alphabetical index, which enables probability mass on codes with few or zero training examples.

(iii) Processing long inputs. BERT-style models necessitate chunking notes due to limited input length (512 tokens), which complicates opti-

[3] *CLH* can, in theory, assign any of the 70K codes. However, this does not imply uniform coverage in practice, since a specific implementation may have 'blind spots' that should be established empirically via performance analysis, as in other classification problems, especially under class-imbalanced.

mization (Pascual et al., 2021). Chunking converts document-level supervision into a multi-instance setting where aggregators such as max-pooling pass gradients only through the highest-scoring segment and cross-segment dependencies are easily missed. Medical notes can be fairly long, up to 8,500 tokens in MIMIC dataset (Johnson et al., 2016, 2023). Although LLMs handle longer inputs better, very long contexts still pose challenges (Karpinska et al., 2024; Li et al., 2024; Kim et al., 2025). Even for most LLMs,[4] providing the entire ICD solution space as in-context retrieval (Lee et al., 2024) is not computationally feasible, as it easily would exceed 1M tokens. It is also not realistic: human coders narrow the solution space using multi-step processes guided by the alphabetical index and official guidelines (Dong et al., 2022; Gan et al., 2025).

Our solution. *CLH* adopts an agentic approach, which decomposes the processing of long context as a sequential task, where the system searches, verifies, and predicts one code at a time based on guidelines. This avoids having to process extremely long context in one go.

3 Background: a Gentle Introduction to Medical Coding

Human medical coders are healthcare professionals with extensive training in one or more medical coding systems.[5]. Given some clinical documentation (progress notes, discharge summaries, etc.), they must choose among the thousands of codes the most specific one that applies to this case. We will use the example of the US modification of ICD-10 (ICD-10-CM) (CMS and NCHS, 2025), which comprises over 70,000 codes. Example snippets from the ICD hierarchy and alphabetical index can be seen in the Appendix A.

These codes are defined in a tree-structured hierarchy where the proximity of the codes indicates some similarity.[6] The alphanumeric characters in a medical code refer to its position in the hierarchy: e.g., in the code *A22.7*, "A" points to a chapter, "22" points to a category, and the numbers after the punctuation indicate the conditions with increasing level of specificity. Only the most specific possible code can be assigned. However, proximity in the hierarchy does not guarantee clinical similarity. For example, 'sepsis' is mentioned 39 times in multiple chapters, and it corresponds to codes *A22.7* and *T81.44*, among many others. Therefore, finding the correct code by traversing from the top of the hierarchy to the bottom is difficult.

To resolve such cases, the coders must also use the alphabetical index, which provides information about the context in which a given code is appropriate. For example, it could help the coder to decide between *A22.7* and *T81.44* codes for 'sepsis', because the former is listed under "anthrax" context, and the latter under "postprocedural".

The alphabetical index alone is also insufficient because it does not necessarily point to the most specific code (i.e., the leaf node in the ICD tree). Hence, the coder must verify that the candidate code is the most specific code possible for this case in the ICD hierarchy.

The output of medical coding is not a single code, but a list of applicable codes. The order of those codes may itself be meaningful in some cases. There are also rules for precedence between specific codes, e.g., the code for "Alzheimer's" should come before the code for "dementia".

All human coders must be familiar with official ICD guidelines to ensure accurate and comprehensive coding. The current ICD-10-CM guidelines are 115 pages long. These guidelines serve as the manual for medical coding and provide instructions on using the alphabetical index and navigating the hierarchy. Additionally, they provide chapter-specific rules on selecting and combining codes, which are fundamental to clarifying otherwise ambiguous choices among several candidate codes.

4 Code Like Humans (*CLH*) Framework

4.1 Architecture

The proposed *CLH* framework decomposes the medical coding task into four steps, corresponding to the *Analyze-Locate-Assign-Verify* approach implemented by the UK National Health Service (NHS England, 2023). We first describe the framework abstractly in terms of the interface and functionality of each component. In section 4.2, we present our implementation, which reflects one of many possible realizations of *CLH* and is not intended as a definitive standard.

[4]Gemini models handle up to 1M tokens (Lee et al., 2024), yet to date most evaluations use the needle-in-the-haystack paradigm (Kamradt, 2023), e.g. (Hsieh et al., 2024), where they perform much better than on a task requiring reasoning over an entire book (Karpinska et al., 2024).

[5]According to Otero Varela et al. (2024), the training for a medical coding certification takes several months.

[6]In ICD, this hierarchy is referred to as the 'tabular list'.

Step 1: *evidence extractor*. This component identifies codeable conditions within clinical notes, surfacing text snippets that may justify codes. [7] A key challenge for this step is that clinical language frequently diverges from standardized index terms, and identifying relevant excerpts is non-trivial.

Step 2: *index navigator*. This component locates the authoritative coding references in the alphabetical index by mapping text snippets to valid index terms (e.g., 'sepsis'). This involves handling synonyms, variant phrasings, and eponyms to propose preliminary candidate codes. The output is a list of candidate codes (e.g. *A22.7*, *T81.44*, etc. for 'sepsis'), each associated with a located term (e.g. 'anthrax' and 'postprocedural sepsis' for the above examples). Since the alphabetical index is not guaranteed to point to an assignable code, the output of this step is only preliminary.

Step 3: *tabular validator*. This component refines and narrows down the candidate codes by applying formal coding rules. It interacts with the ICD hierarchy and chapter-specific guidelines to resolve ambiguities and anatomical specifications, thereby producing a tentative code set.

Step 4: *code reconciler*. This component finalizes the code assignment, applying instructional notes to resolve mutually exclusive codes and ordering conventions. The output is intended to be the most complete and ordered list of codes that reflects the patient encounter, while adhering to medical coding conventions.

4.2 Implementation

We implement the *CLH* framework as an agentic system (see Figure 1). We define agents as modular components that can be implemented in various ways, such as standalone language models and retrieval-augmented generation (RAG).

In our implementation, each agent is instantiated as a distinct inference step using the same backbone model, but with different instructions (via prompts) tailored to their role (see Appendix C). The current implementation relies on models with 'thinking-enabled' mode to enable test time computation (Snell et al., 2024; Muennighoff et al., 2025).

We use the 'reasoning' models for improved performance; theoretically the 'reasoning traces' could also provide transparency into the model decision process, but at least for the current 'reasoning' models this process is not faithful (Chen et al., 2025; Kambhampati et al., 2025; Shojaee et al., 2025; Zhao et al., 2025). Specifically, we experiment with three open-weight models of different sizes: *small* (DeepSeek-R1-0528-Qwen3-8B (DeepSeek-AI et al., 2025)), *base* (DeepSeek-R1-Distill-LLaMA-70B (DeepSeek-AI et al., 2025)), and *large* (Qwen3-235B-A22B (Team Qwen, 2025)). These models were run locally, ensuring HIPAA compliance. Additionally, we evaluate o3-mini (OpenAI, 2025a) and o4-mini (OpenAI, 2025b), under a HIPAA-compliant use setup. Further implementation details are provided in Appendix B.

The *evidence extractor (step 1)* operates solely on clinical notes, extracting verbatim text snippets expected to justify coding decisions. For each snippet, we retrieve the top-10 alphabetical terms by embedding snippets and terms into the same semantic space (see appendix D.1). Next, the *index navigator (step 2)* processes each snippet's set of terms in parallel, selecting the most appropriate terms, which in turn yield a set of candidate codes. These codes, often grouped by ICD chapter, are then passed in parallel to the *tabular validator (step 3)* along with chapter-level guidelines,[8] receiving input triplets of {clinical note, chapter guidelines, candidate codes} to yield tentative assigned codes. Finally, outputs from parallel processing steps are merged, and the *code reconciler (step 4)* receives input triplets of {clinical note, instructional notes, tentative codes} to verify and finalize the coding assignments, where instructional notes are retrieved by looking up codes in the ICD hierarchy.

4.3 Medical code taxonomy

We focus on ICD-10-CM, as (i) ICD-10 is the most widely adopted coding system globally (Teng et al., 2023), (ii) its introduction made the coding task much more laborious and time-consuming,[9] as the number of codes increased more than sevenfold (rising from 9,000 to 70,000), and (iii) it is well-maintained with high-quality open-access

[7] One challenge is that in US outpatient care (but not inpatient care) notes may mention suspected rather than diagnosed conditions, which should be ignored for coding. Similarly, similar consequential rules may exist in other settings, further underscoring the task's complexity (CMS and NCHS, 2025).

[8] The guidelines span 100+ pages; each chapter averages about three pages. Parallel processing lets us input only the chapters relevant to the candidate codes, typically one or two.

[9] It takes a professional inpatient human coder approximately 30 minutes per case on average (Chen et al., 2021).

resources to guide coding decisions.

4.4 Data

MIMIC is the most widely used open-access database for research on medical coding. We use the popular MIMIC-III *50* split (Mullenbach et al., 2018) for fine-tuning.

MDACE (Cheng et al., 2023) is a dataset of 4,000 human-verified annotations that link ICD codes to supporting evidence spans within clinical notes. The annotations cover 302 inpatient charts and 52 professional-fee charts from MIMIC-III (Johnson et al., 2016), including discharge summaries, physician notes, radiology reports, and other clinical document types representative of real-world coding contexts (Alonso et al., 2020). Inter-annotator agreement after adjudication is high (Krippendorff's $\alpha = 0.97$ for inpatient; 0.96 for Profee). We use MDACE for evaluation because it includes human-verified evidence spans ideal for LLM evaluation and directly supports ICD-10-CM, our target taxonomy. To our best knowledge, MDACE is the only public dataset that addresses two well-known issues in MIMIC-based benchmarks: the absence of code-to-text links and broader validity concerns about treating MIMIC codes as a gold standard (Searle et al., 2020; Kim and Ganapathi, 2021). To our knowledge, MDACE is the first and remains among the very few public resources with token-level evidence for long clinical notes in extreme multi-label coding.

MIMIC and MDACE come with MIT license. The MIMIC data is distributed by PhysioNet, which requires HIPAA compliance.

4.5 Evaluation metrics

Like previous studies, we report F1 scores (micro and macro), exact match ratio (EMR), and recall for each agent. We pay particular attention to macro F1, which assigns equal weight to every label. Edin et al. (2023) shows that F1 macro is shaped by the long tail of low-frequency labels, where per-label F1 increases with log frequency until roughly 100 training examples and then plateaus. We call this long-tail subset "rare codes"; gains on rare codes, therefore, translate directly into higher F1 macro.

4.6 Baselines

PLM-ICD. As discussed in section 2, the state-of-the-art system for this task is still the PLM-ICD (Huang et al., 2022). We follow the implementation details by Edin et al. (2024) to reproduce the model and evaluate it on the test split of MDACE.

LLM out-of-the-box. We also compare the *CLH* approach to the 'naive' LLM-based solution: the DeepSeek R1 distilled Llama3.3-70B that is provided with the set of possible codes in the prompt. Following the PLM-ICD setup, in this experiment, the model is provided with only the subset that occurs in the MDACE dataset and not the full set of ICD-10 codes.

5 Results

5.1 End-to-end evaluation

We evaluated our implementation of the *CLH* framework end-to-end, comparing it directly to baselines on MDACE. Table 1 summarizes the results. With the largest base model, *CLH* achieves comparable performance with PLM-ICD, while handling a 70-fold larger label space.

Notably, while PLM-ICD achieves higher performance on frequent codes (as indicated by its superior F1 micro score), this advantage primarily stems from its supervised training regime, which utilizes data from the same intensive care unit (ICU) as the test set. This shared setting creates strong distributional priors, allowing PLM-ICD to excel specifically on frequent codes within that clinical practice. Conversely, *CLH* does not benefit from such distributional priors, which explains its superior performance on rare codes.

To our knowledge, no prior work has considered the full label space. Hence, when considering the realistic task of human coders, our results can be regarded as the state-of-the-art. In the future, it may be helpful and fairer to compare medical coding systems in the full and label-constrained settings.

5.2 Performance of the *CLH* agents

As discussed in subsection 4.1, the rationale for introducing the *CLH* framework is the 4-step process used by medical coders at the NHS. To consider whether this process was indeed beneficial to *CLH*, Table 2 reports the results for each component in the *filtered* setting (i.e., the errors made at the previous step of the pipeline are discarded).

We observe that the *index navigator (step 2)* achieves high recall with low precision, suggesting broad coverage of candidate codes. However, the *tabular validator (step 3)* and the *code reconciler (step 4)* show much higher precision while maintaining high recall.

Model	#P	#C	F1 Micro	F1 Macro	EMR
constrained label space (prior work)					
PLM-ICD	340M	$1K$	**0.48**	0.25	0.02
PLM-ICD	340M	$6K$	0.46	0.21	0.02
Llama3-70B†	70B	$1K$	0.28	0.18	0.01
CLH-small	8B	$1K$	0.27	0.18	0.02
CLH-base	70B	$1K$	0.38	0.24	0.02
CLH-large	235B	$1K$	0.43	**0.28**	0.02
CLH-o3-mini	–	$1K$	0.37	0.24	0.02
CLH-o4-mini	–	$1K$	0.41	0.27	0.02
full label space (ours, realistic clinical setting)					
CLH-base	70B	$70K$	0.32	0.14	0.02

† The DeepSeek-R1 distilled Llama3-70B model.

Table 1: Performance of the *CLH* framework and baselines on the MDACE dataset. "#P" refers to the number of parameters and "#C" to the number of candidate codes used during inference. For PLM-ICD, this includes all MIMIC codes ($\approx 6K$) which overlap with MDACE. Llama3 is prompted with all MDACE codes ($\approx 1K$). *CLH* scales to the full ICD coding system ($\approx 70K$), but is also evaluated in the MDACE constrained 1K setting for comparison.

As with any pipeline approach, *CLH* can propagate early mistakes to later agents. A promising direction for mitigating this in future work is self-refinement (Madaan et al., 2023) on the same clinical note. After pass t, the code reconciler (step 4) outputs a set of codes with rationales. We could append this output to the note as a lightweight scratchpad, then re-invoke all steps once again. The process then repeats for $t{+}1$.

6 Model Analysis

While the end-to-end results provide a summary, they do not capture the nuances of the individual stages of the framework. To better understand the limitations of *CLH*, we present five observations to guide future improvements.

6.1 Are all code-able cues identified?

Falis et al. (2024) observed that GPT-3.5 works well for coding only when code descriptions are verbatim. That setting is rare in practice, where cues embed abbreviations, medical jargon, and implied context. We analyzed the MDACE expert-annotated evidence spans with the text snippets extracted by the *index navigator (step 2)*. Then, we measured recall at the level of ICD-10 classification chapters to identify which cues the agent overlooks, and which the retriever fails to link to alphabetical-index terms.

Model	F1 Micro	F1 Macro	Recall	Precision
1. evidence extractor	0.12	0.09	0.62	0.06
2. index navigator	0.36	0.25	0.53	0.27
3. tabular validator	0.40	0.27	0.47	0.34
4. code reconciler	0.43	0.28	0.46	0.36

Table 2: Performance of the individual components of the *CLH-large* framework on MDACE dataset. Results reflect progressive improvements through the pipeline.

Figure 2 shows the retrieval recall@25 for different ICD-10 chapters. X-axis shows recall based on alphabetical terms retrieved using the *evidence extractor (step 1)*, while the y-axis reports recall based on expert-annotated text snippets. In each case, we calculate recall for the top 25 retrieved ICD-10 terms. High recall for J00–J99 (respiratory) and N00–N99 (genitourinary) shows that both the agent and expert annotations reliably retrieve index terms for body-system diseases. When the expert annotations recall is limited, as for C00–D49 (neoplasms), S00–T88 (injury or poisoning), and V00–Y99 (external causes), the agent cannot compensate, indicating a retrieval bottleneck. In contrast, we find that for the chapters F01–F99 (mental and behavioral disorders), H00–H59 (eye and adnexa), and Z00–Z99 (factors influencing health and contact with services), the *evidence extractor* agent underperforms significantly.

To further investigate this shortcoming, we inspected the word clouds for the evidence spans linked to false negatives (see Figure 8 in the subsection D.2). Frequent misses involve abbreviations (*Hx of CVA*), social or historical qualifiers (*quit smoking*, *tobacco use*), administrative directives (*DNR*), and medication names (*warfarin*, *coumadin*). These cues are not active disease mentions but modifiers that require abbreviation resolution, temporal inference, and context classification. Psychological terms such as *depression* and *anxiety* are likewise overlooked, explaining the agent's low performance on F01–F99.

Given that much note content is noise (Liu et al., 2022), we speculate that *CLH* could operate on entity+assertion spans rather than full notes, a choice recently shown to maintain accuracy and provide clearer entity-level evidence (Douglas et al., 2025).

6.2 What is the impact of more candidate codes?

In this section, we study the effect of the candidate space in a controlled setting that does not rely on

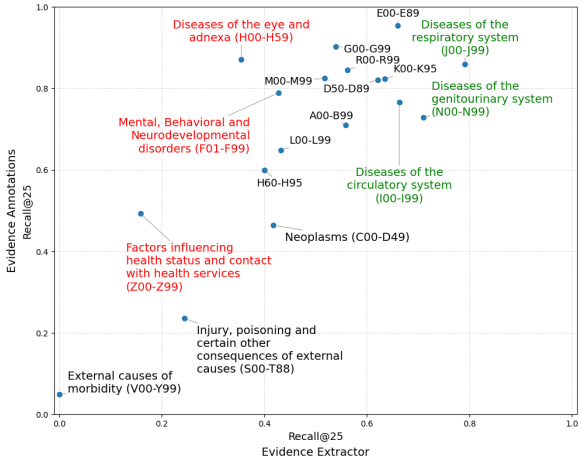

Figure 2: Chapter-level comparison of retrieval recall@25. The x-axis reports retrieving alphabetical index terms with the *1. evidence extractor*, while the y-axis reports retrieval with expert-annotated evidence.

CLH. At inference time we construct a per-note *candidate set* C by combining the ground-truth codes $P = \{y_i : y_i = 1\}$ with K hard negatives per positive: $C = P \cup R_K$ with $|R_K| = K|P|$. We identify hard negatives by embedding the short descriptions of codes using our retriever (*S-PubMedBert-MS-MARCO*, see subsection D.1) and sampling the K nearest neighbors that are not in P. This allows us to examine how the size of the candidate set affects model behavior. We then run the *tabular validator (step 3)* and the *code reconciler (step 4)* only on the candidate set C. The *tabular validator (step 3)* receives mutually exclusive choices, which simplifies selection, while the *code reconciler (step 4)* operates in a multi-label setting and must also decide how many codes to output.

Figure 3 shows the F1 micro scores of the *tabular validator (step 3)* and *code reconciler (step 4)* when scaling the number of negative codes per positive code. As the candidate set grows, both agents exhibit a decline in performance, confirming the challenges of reasoning over larger solution spaces in long, context-rich sequences (Li et al., 2024). However, the *tabular validator (step 3)* proves more robust, benefiting from a mutually exclusive candidate set that simplifies the selection task. In contrast, the *code reconciler (step 4)* must also determine the number of correct codes to output, compounding the difficulty in a multi-label setting. These findings validate our framework's modular design, where the *tabular validator (step 3)* predicts tentative codes in parallel, and the *code reconciler (step 4)* audits them in a final step.

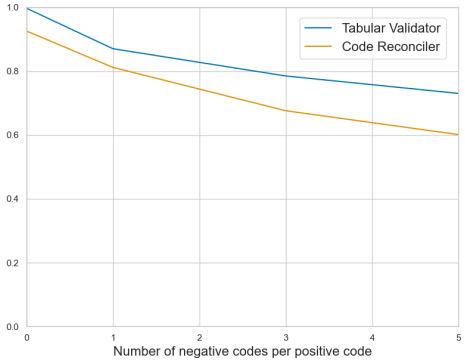

Figure 3: F1 micro scores for the *tabular validator (step 3)* and *code reconciler (step 4)* as the number of negative codes increases. The *code reconciler (step 4)* remains more robust in larger candidate sets due to its simpler prediction task.

6.3 Do the ICD-10 coding resources help?

Though the context window of some LLMs exceeds millions of tokens (Gemini Team et al., 2024), they are known to have an *effective* context window of an arbitrary smaller size (Lee et al., 2024). We conducted an ablation study to establish how the capacity of *code reconciler (step 4)* saturates with adding more data. We incrementally enriched the input with code-specific information to examine how the model leverages the added context. We started with (1) alphanumeric code identifiers, adding (2) short code descriptions, and then (3) chapter-specific guidelines. We follow the same setup as in subsection 6.2 to construct the candidate set.

Figure 4 shows that enriching the context input with structured guideline information affects the performance of *tabular validator (step 3)*. Starting with only alphanumeric code identifiers, we observe that adding short descriptions leads to clear improvements. Including chapter-specific coding guidelines yields further gains, particularly as the number of candidate codes increases. These results highlight two observations: the model benefits from in-context reasoning grounded in guidelines, and the ability to process a larger candidate set improves with access to richer contextual information about codes. However, as the trend is unlikely to continue infinitely, in practice, the effective size of the context should be identified for a given model.

6.4 Does 'thinking-enabled' decoding help?

Test-time compute has recently improved accuracy on multiple-choice QA and math benchmarks (Snell et al., 2024; Muennighoff et al., 2025). We analyzed the performance impact

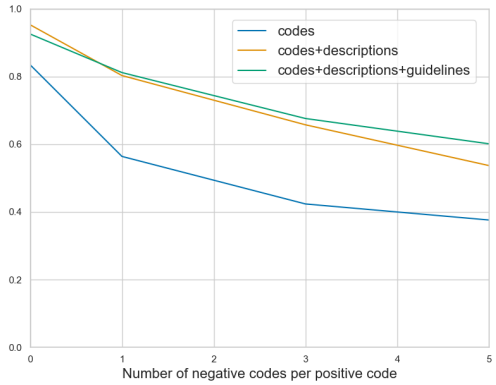

Figure 4: F1 micro scores for *3. tabular validator* with added context. Guideline information yields the strongest gains as candidate codes increase.

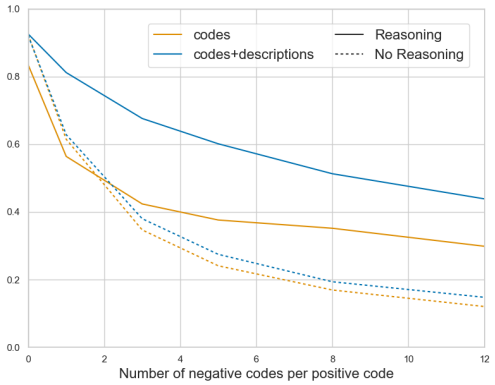

Figure 5: F1 micro scores for the *3. tabular validator* with and without reasoning. Reasoning is consistently superior, especially with more negative codes and longer contexts.

of 'thinking-enabled' output generation employing the *3. tabular validator*. This setup begins with an unconstrained 'reasoning' phase, enclosed within `<think>...</think>` tags, followed by a constrained output phase within `<answer>...</answer>` tags. We compared this to structured decoding, where outputs are constrained using a regex pattern to a predefined set of candidate codes (Willard and Louf, 2023), and code prediction begins from the outset of the generation.

Figure 5 compares F1 micro score as the number of negative codes per positive code increases for the *3. tabular validator* with and without 'thinking-enabled' generation. 'Thinking-enabled' generation consistently outperforms structured decoding. Both settings achieve near-perfect accuracy at zero negatives, yet the gap widens as the candidate set and context length grow. The gentler slope suggests the agent can better explore and exploit in-context information throughout the generation process. In contrast, the structured decoding approach initiates code prediction from the outset, limiting the benefit from additional context. It only improves performance marginally when the alphanumeric codes are extended with semantic descriptions.

In the scope of this work, we consider primarily the performance of the 'thinking' mode, and do not investigate its interpretability or faithfulness.

6.5 Can fine-tuning work as well for LLMs as for BERT?

Despite the growing interest in applying auto-regressive LLMs to clinical text, to our knowledge, no study has fine-tuned them for medical coding. We choose `Llama-3.2-1B` as our backbone model, and fine-tuned it using LoRA adapters on all the projection matrices in the attention layer and feed-forward network. We did not apply any adapters to the language-modeling head. We use supervised fine-tuning, disregarding the order of the codes generated by the model. Due to the limitation of the input size of this model, we can only provide 50 code options in the model prompt. We train and test on MIMIC-III-50, as this allows us to provide descriptions for all 50 codes in the prompt. For efficiency reasons, we subset the training, validation, and test sets to 7500, 2000, and 2000 examples. We train with an effective batch size of 32 for five epochs. We pick the model associated with the best validation F1 macro to produce our test-set metrics (see Appendix E for more details). Table 3 shows that fine-tuning a Llama-3.2-1B, a generative model, for medical coding severely underperforms PLM-ICD (a fine-tuned model of BERT generation).

We speculate that much of PLM-ICD's advantage over predecessor models is due to label-wise attention. We repeat our previous experiment to investigate this empirically, but replace the causal language modeling head with label-wise attention. We follow Edin et al. (2024), implementing it as cross-attention between all tokens and a learned representation per code. We fully train this in addition to the LoRA updates described above. This changes the task from auto-regressive generation to predicting probabilities over the entire label space simultaneously. We tune the decision boundary based on the highest F1 macro score. We use the boundary associated with the best validation set performance for our test scores. All other fine-tuning details are identical to the auto-regressive setting.

Table 3 shows that the performance gap is closed

	F1	
Model	**Micro**	**Macro**
PLM-ICD	0.71	0.66
Llama-3.2-1B	0.40	0.24
+ Label-wise attention	0.71	0.65

Table 3: F1 scores on MIMIC-III-50 for PLM-ICD, fine-tuned Llama-3.2-1B without and with label-wise attention. PLM-ICD scores from Edin et al. (2023).

when label-wise attention is added to the backbone model, and the performance matches PLM-ICD. It is interesting that even with the label-wise attention, the `Llama-3.2-1B` model only manages to recover the original performance of MIMIC-III-50 from Edin et al. (2023), despite being 9 times larger.

7 Discussion

7.1 Are LLMs ready for clinical practice?

Discriminative classifiers, such as PLM-ICD, retain knowledge about every code in their weights, whereas our *CLH* framework dynamically retrieves ICD codes from the alphabetic index, enabling true open-set coding at inference. *CLH* does not yet outperform the PLM-ICD baseline on frequent codes, but it achieves comparable performance, and has several practical advantages. *CLH* is much easier to update when coding conventions change: one would only need to substitute external resources in the prompt. It also suffers less from distributional priors and theoretically can handle even extremely rare codes such as *W59.22 "Struck by a turtle"* (Edin et al., 2023).

Nevertheless, both discriminative classifiers and LLMs have apparent limitations that would make both approaches unusable for the real-world ICD-10 end-to-end classification. The former inherit strong distributional priors and only support a toy version of the real 70K label task. The latter still struggles with precision. Given the current state of the technology, we argue that a more realistic goal for NLP for medical coding is to develop assistive tools that keep human coders in the loop, rather than aim for full end-to-end classification. In that approach, *CLH* has a key advantage, as it already mimics the workflow of human coders.

7.2 Future work

Human-computer interaction. *CLH* framework provides an exciting opportunity to reimagine the task of medical coding as computer-assisted coding, where the goal is not to automate the task but to provide the human coder with the best information to base their decisions on (Murray, 2024). This could involve research into user interfaces, identification of the pain points where the coders actually want assistance, the threshold of information overload, potential sources of automation bias, and strategies for mitigating them.

Improving *CLH* components. *CLH* allows for more tailored instruction fine-tuning strategies for each component of the pipeline. For example, the *evidence extractor (step 1)* can be tuned to predict span-level clinical entities with assertion status and key modifiers (e.g., laterality, acuity) (Douglas et al., 2025), and to pass only asserted, coding-eligible evidence forward, which reduces noise while preserving code-relevant detail. The *index navigator (step 2)* can then align entity representations to alphabetical index terms using a label-wise cross-attention objective that improves handling of synonyms and eponyms. The *tabular validator (step 3)* could be fine-tuned using clinical notes paired with correct codes and closely related incorrect codes, enabling the model to leverage external resources to identify the most appropriate code. Finally, the *code reconciler (step 4)* can be fine-tuned using clinical notes paired with candidate code lists, incorporating hard negatives to train the model in distinguishing relevant from irrelevant codes with greater precision.

8 Conclusion

We introduced *Code Like Humans*, a multi-agent framework for medical coding that aligns with human coding practices by integrating previously overlooked resources: the alphabetical index and official guidelines of ICD-10 classification. By relying on these in-context materials rather than only training examples, CLH became the first solution that could support the complete ICD-10 system (over 70K codes). The current implementation outperforms state-of-the-art fine-tuned classifiers on rare codes. A performance gap remains for frequent codes, where fine-tuned classifiers have a natural advantage, but this comes at the expense of addressing only a toy version of the real ICD-10 coding task.

Our framework offers advantages in easy adaptability to yearly coding updates and opportunities for human-in-the-loop collaboration. Our results suggest that while the current NLP systems can support medical coding, they should aim to complement rather than replace human expertise.

Limitations

Variety of medical coding practices. Medical coding varies widely in complexity depending on the coding system (e.g., ICD-9, ICD-10, CPT, etc.), clinical setting (e.g., inpatient, outpatient, professional fee), and document type (e.g., progress notes, radiology reports, discharge summaries). In the scope of this work, we only consider the case of ICD-10 classification, and our evaluation is bounded by the cases considered in the MDACE dataset. More studies considering various models and classification schemes, as well as more public medical benchmark datasets, are needed to address the needs of different healthcare providers.

How effective is coverage of ICD-10? Ours is the first work to support the full ICD-10 label space. Still, the current evaluation does not allow for a comprehensive description of how the system would perform in various parts of the taxonomy. The core issue is that there are currently no public evaluation resources that cover the full label space (which is why all prior work has so far focused on a small subset of the codes). This makes it impossible to evaluate any system for performance on all the cases that could be encountered in clinical practice. Development of resources that cover ICD-10 is crucial for further progress in this area.

Supported languages. Like most other work on clinical NLP, this work focuses on English. There is a growing need for NLP systems to support healthcare assistance in languages other than English, as well as for other medical coding principles and practices worldwide.

Evaluation caveats. PLM-ICD was fine-tuned on MIMIC-IV ICD-10. Since MDACE contains notes from the same hospital, this could provide an unfair advantage to PLM-ICD over the other models. Conversely, the MIMIC-IV annotations are noisy, where many annotated codes are never mentioned in the clinical notes (Cheng et al., 2023). The comparison between PLM-ICD and the other models might have been different if PLM-ICD had been trained on cleaner data from another hospital.

The clinical notes in MDACE and MIMIC have been anonymized (i.e., the special characters have replaced names, dates, addresses, and other identifiers). These special characters may be a source of additional errors for the models.

As elaborated in subsection 7.2, we encourage future work on improving *CLH*, and experimenting with other models is a natural direction to pursue.

Broader Impacts

Regulation of high-risk AI applications. According to the enacted EU AI Act (European Union, 2024), the list of high-risk applications includes "AI systems intended to be used by public authorities or on behalf of public authorities to evaluate the eligibility of natural persons for essential public assistance benefits and services, including healthcare services, as well as to grant, reduce, revoke, or reclaim such benefits and services". Where automated medical coding informs eligibility assessment for healthcare services or benefits, it could constitute a high-risk system under the EU AI Act, and carry significant risks to patient care and revenue cycles.

In subsection 7.1, we argued that at the current state of LLM technology, a more realistic and safe approach is to develop methods that assist medical coders rather than provide automated recommendations, while also mitigating the potential automation biases that an assisted workflow may introduce.

Privacy risks. All work reported in this study was performed using publicly available models and medical datasets, under their existing licenses. We created no new data or base models that could pose new privacy risks.

***CLH* availability.** Our implementation of *CLH* is publicly released under the MIT license, facilitating further research and development. Our solution is intended and marked as a research prototype, not as something that can be deployed out of the box for medical coding applications.

Acknowledgments

Generative coding assistance (GitHub CoPilot and ChatGPT) was used in the development of *CLH* implementation.

This work was supported by the Innovation Fund Denmark and Corti ApS through the Industrial PhD program (grant no. 4297-00057B) of Andreas Geert Motzfeldt at the IT University of Copenhagen.

References

Vera Alonso, João Vasco Santos, Marta Pinto, Joana Ferreira, Isabel Lema, Fernando Lopes, and Alberto

Freitas. 2020. Problems and Barriers during the Process of Clinical Coding: a Focus Group Study of Coders' Perceptions. *Journal of Medical Systems*, 44(3):62.

Joseph S. Boyle, Antanas Kascenas, Pat Lok, Maria Liakata, and Alison Q. O'Neil. 2023. Automated clinical coding using off-the-shelf large language models. In *Proceedings of the NeurIPS 2023 Workshop on Deep Generative Models for Health (DGM4H)*. Poster.

Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. 2020. HyperCore: Hyperbolic and co-graph representation for automatic ICD coding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3105–3114, Online. Association for Computational Linguistics.

Rajiv Chandawarkar, Prakash Nadkarni, Elizabeth Barmash, Stephany Thomas, Allison Capek, Kathleen Casey, and Freddy Carradero. 2024. Revenue Cycle Management: The Art and the Science. *Plastic and Reconstructive Surgery Global Open*, 12(7):e5756.

Pei-Fu Chen, Ssu-Ming Wang, Wei-Chih Liao, Lu-Cheng Kuo, Kuan-Chih Chen, Yu-Cheng Lin, Chi-Yu Yang, Chi-Hao Chiu, Shu-Chih Chang, and Feipei Lai. 2021. Automatic ICD-10 Coding and Training System: Deep Neural Network Based on Supervised Learning. *JMIR Medical Informatics*, 9(8):e23230. Company: JMIR Medical Informatics Distributor: JMIR Medical Informatics Institution: JMIR Medical Informatics Label: JMIR Medical Informatics Publisher: JMIR Publications Inc., Toronto, Canada.

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, Vlad Mikulik, Samuel R. Bowman, Jan Leike, Jared Kaplan, and Ethan Perez. 2025. Reasoning Models Don't Always Say What They Think.

Hua Cheng, Rana Jafari, April Russell, Russell Klopfer, Edmond Lu, Benjamin Striner, and Matthew 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, Toronto, Canada. Association for Computational Linguistics.

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).

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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?

Hang Dong, Víctor Suárez-Paniagua, William Whiteley, and Honghan Wu. 2020. Explainable Automated Coding of Clinical Notes using Hierarchical Label-wise Attention Networks and Label Embedding Initialisation.

James C. Douglas, Yidong Gan, Ben Hachey, and Jonathan K. Kummerfeld. 2025. Less is More: Explainable and Efficient ICD Code Prediction with Clinical Entities. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30835–30847, Vienna, Austria. Association for Computational Linguistics.

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, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 2572–2582. ACM.

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.

European Union. 2024. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act) (Text with EEA relevance).

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

Adam Gaffney, Stephanie Woolhandler, Christopher Cai, David Bor, Jessica Himmelstein, Danny McCormick, and David U. Himmelstein. 2022. Medical Documentation Burden Among US Office-Based Physicians in 2019: A National Study. *JAMA Internal Medicine*, 182(5):564–566.

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.

Yue Gao, Yuepeng Chen, Minghao Wang, Jinge Wu, Yunsoo Kim, Kaiyin Zhou, Miao Li, Xien Liu, Xiangling Fu, Ji Wu, and Honghan Wu. 2024. Optimising the paradigms of human AI collaborative clinical coding. *npj Digital Medicine*, 7(1):1–15. Publisher: Nature Publishing Group.

Google Gemini Team, Quoc Le, Arjun Kar, Madhu Gurumurthy, Cheng Li, Ruoxin Sang, Fangyu Liu, Lampros Lamprou, Rich Munoz, Nathan Lintz, Harsh Mehta, Heidi Howard, Malcolm Reynolds, Lora Aroyo, Quan Wang, Lorenzo Blanco, Albin Cassirer, Jordan Griffith, Dipanjan Das, and 932 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context.

Zelalem Gero, Chandan Singh, Hao Cheng, Tristan Naumann, Michel Galley, Jianfeng Gao, and Hoifung Poon. 2023. Self-Verification Improves Few-Shot Clinical Information Extraction.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. 2024. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*.

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 E. W. Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J. Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, Liwei H. Lehman, Leo A. Celi, and Roger G. Mark. 2023. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific Data*, 10(1):1. Publisher: Nature Publishing Group.

Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Liwei 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):160035. Publisher: Nature Publishing Group.

Subbarao Kambhampati, Kaya Stechly, Karthik Valmeekam, Lucas Saldyt, Siddhant Bhambri, Vardhan Palod, Atharva Gundawar, Soumya Rani Samineni, Durgesh Kalwar, and Upasana Biswas. 2025. Stop Anthropomorphizing Intermediate Tokens as Reasoning/Thinking Traces!

Gregory Kamradt. 2023. Needle in a haystack - pressure testing llms. GitHub Repository.

Marzena Karpinska, Katherine Thai, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. One Thousand and One Pairs: A "novel" challenge for long-context language models.

Byung-Hak Kim and Varun Ganapathi. 2021. Read, Attend, and Code: Pushing the Limits of Medical Codes Prediction from Clinical Notes by Machines. In *Proceedings of the 6th Machine Learning for Healthcare Conference*, pages 196–208. PMLR. ISSN: 2640-3498.

Yekyung Kim, Jenna Russell, Marzena Karpinska, and Mohit Iyyer. 2025. One ruler to measure them all: Benchmarking multilingual long-context language models.

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.

Jinhyuk Lee, Anthony Chen, Zhuyun Dai, Dheeru Dua, Devendra Singh Sachan, Michael Boratko, Yi Luan, Sébastien M. R. Arnold, Vincent Perot, Siddharth Dalmia, Hexiang Hu, Xudong Lin, Panupong Pasupat, Aida Amini, Jeremy R. Cole, Sebastian Riedel, Iftekhar Naim, Ming-Wei Chang, and Kelvin Guu. 2024. Can Long-Context Language Models Subsume Retrieval, RAG, SQL, and More?

Fei Li and Hong Yu. 2020. ICD coding from clinical text using multi-filter residual convolutional neural network. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8180–8187. AAAI Press.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024. Long-context LLMs Struggle with Long In-context Learning.

Jinghui Liu, Daniel Capurro, Anthony Nguyen, and Karin Verspoor. 2022. "Note Bloat" impacts deep learning-based NLP models for clinical prediction tasks. *Journal of Biomedical Informatics*, 133:104149.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder,

Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling.

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.

Alex Murray. 2024. Daron Acemoglu and Simon Johnson. Power and Progress: Our Thousand-Year Struggle Over Technology and Prosperity. *Administrative Science Quarterly*, 69(4):NP84–NP87. Publisher: SAGE Publications Inc.

NHS England. 2023. National Clinical Coding Standards ICD-10 5th Edition (2023). Technical report, Terminology and Classifications Delivery Service.

OpenAI. 2025a. Openai o3-mini. Technical Report.

OpenAI. 2025b. Introducing openai o3 and o4-mini. Technical Report.

Lucia Otero Varela, Chelsea Doktorchik, Natalie Wiebe, Danielle A. Southern, Søren Knudsen, Pallavi Mathur, Hude Quan, and Cathy A. Eastwood. 2024. International Classification of Diseases clinical coding training: An international survey. *Health Information Management: Journal of the Health Information Management Association of Australia*, 53(2):68–75.

Damian Pascual, Sandro Luck, and Roger Wattenhofer. 2021. Towards BERT-based automatic ICD coding: Limitations and opportunities. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 54–63, Online. Association for Computational Linguistics.

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

Thomas Searle, Zina Ibrahim, and Richard Dobson. 2020. Experimental evaluation and development of a silver-standard for the MIMIC-III clinical coding dataset. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pages 76–85, Online. Association for Computational Linguistics.

Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. 2025. The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters.

Team Qwen. 2025. Qwen3: Think Deeper, Act Faster. Section: blog.

Fei Teng, Yiming Liu, Tianrui Li, Yi Zhang, Shuangqing Li, and Yue Zhao. 2023. A Review on Deep Neural Networks for ICD Coding. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4357–4375. Conference Name: IEEE Transactions on Knowledge and Data Engineering.

Thanh Vu, Dat Quoc Nguyen, and Anthony Nguyen. 2020. A label attention model for ICD coding from clinical text. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3335–3341. ijcai.org.

Brandon T. Willard and Rémi Louf. 2023. Efficient Guided Generation for Large Language Models.

Xiancheng Xie, Yun Xiong, Philip S. Yu, and Yangyong Zhu. 2019. EHR coding with multi-scale feature attention and structured knowledge graph propagation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 649–658. ACM.

Zhichao Yang, Sunjae Kwon, Zonghai Yao, and Hong Yu. 2023. Multi-label few-shot ICD coding as autoregressive generation with prompt. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 5366–5374. AAAI Press.

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.

Chengshuai Zhao, Zhen Tan, Pingchuan Ma, Dawei Li, Bohan Jiang, Yancheng Wang, Yingzhen Yang, and Huan Liu. 2025. Is chain-of-thought reasoning of llms a mirage? a data distribution lens.

A External modules

Here we show the structure of the ICD alphabetic index and hierarchy. The alphabetic index provides an entry point for locating terms and conditions, organized alphabetically and supplemented by subterms and cross-references where each guide term is associated with a code. The ICD hierarchy presents the codes in a hierarchical, chapter-based format organized by body systems or types of conditions. Figure 6 and Figure 7 show representative excerpts from both components to support understanding of the external modules leveraged in our framework.

Sepsis (generalized) (unspecified organism) A41.9
- with
- - organ dysfunction (acute) (multiple) R65.20
- - - with septic shock R65.21
- Acinetobacter baumannii A41.54
- actinomycotic A42.7
- adrenal hemorrhage syndrome (meningococcal) A39.1
- anaerobic A41.4
- Bacillus anthracis A22.7
- Brucella *-see also* Brucellosis A23.9
- candidal B37.7
- Cronobacter A41.59
- cryptogenic A41.9
- due to device, implant or graft T85.79
- - arterial graft NEC T82.7
- - breast (implant) T85.79
- - catheter NEC T85.79
- - - dialysis (renal) T82.7
- - - - intraperitoneal T85.71
- - - infusion NEC T82.7
- - - - spinal (cranial) (epidural) (intrathecal) (spinal) (subarachnoid) (subdural) T85.735

Figure 6: Excerpt from the ICD-10 alphabetical index highlighting the entry term “Sepsis" and indented subterms that guide human coders to the correct term.

A.1 ICD Hierarchy

Chapter 1

Certain infectious and parasitic diseases (A00-B99)

Includes: diseases generally recognized as communicable or transmissible

Use additional code to identify resistance to antimicrobial drugs (Z16.-)

Excludes1: certain localized infections - see body system-related chapters

Excludes2: carrier or suspected carrier of infectious disease (Z22.-)
infectious and parasitic diseases complicating pregnancy, childbirth and the puerperium (O98.-)
infectious and parasitic diseases specific to the perinatal period (P35-P39)
influenza and other acute respiratory infections (J00-J22)

This chapter contains the following blocks:

A00-A09	Intestinal infectious diseases
A15-A19	Tuberculosis
A20-A28	Certain zoonotic bacterial diseases
A30-A49	Other bacterial diseases
A50-A64	Infections with a predominantly sexual mode of transmission
A65-A69	Other spirochetal diseases
A70-A74	Other diseases caused by chlamydiae
A75-A79	Rickettsioses
A80-A89	Viral and prion infections of the central nervous system
A90-A99	Arthropod-borne viral fevers and viral hemorrhagic fevers
B00-B09	Viral infections characterized by skin and mucous membrane lesions
B10	Other human herpesviruses
B15-B19	Viral hepatitis
B20	Human immunodeficiency virus [HIV] disease
B25-B34	Other viral diseases
B35-B49	Mycoses
B50-B64	Protozoal diseases
B65-B83	Helminthiases
B85-B89	Pediculosis, acariasis and other infestations
B90-B94	Sequelae of infectious and parasitic diseases
B95-B97	Bacterial and viral infectious agents
B99	Other infectious diseases

Intestinal infectious diseases (A00-A09)

A00 Cholera

A00.0 Cholera due to Vibrio cholerae 01, biovar cholerae
Classical cholera

A00.1 Cholera due to Vibrio cholerae 01, biovar eltor
Cholera eltor

A00.9 Cholera, unspecified

A01 Typhoid and paratyphoid fevers

A01.0 Typhoid fever
Infection due to Salmonella typhi

A01.00 Typhoid fever, unspecified

A01.01 Typhoid meningitis

A01.02 Typhoid fever with heart involvement

Figure 7: Excerpt from the ICD-10-CM hierarchy (tabular index) showing the beginning of *Chapter 1: Certain infectious and parasitic diseases (A00–B99)*, including instructional notes (includes, use additional, excludes, etc.), followed by hierarchical code blocks and detailed diagnostic categories.

B Implementation details

Our framework implementation is built on the open-source vLLM inference engine, which enables high-throughput, low-latency serving of LLMs by employing PagedAttention and continuous batching (Kwon et al., 2023). vLLM allows us to support long-context inference and concurrent processing efficiently. We power our implementation using four NVIDIA A100 GPUs, each with 80GB of VRAM.

C Prompts

All prompts used in this study were employed with Jinja2-compliant YAML files. This format enables flexible and dynamic prompt construction through template variables. The prompt files are compatible with both completion and chat completion endpoints.

Evidence Extractor

```
You are a highly skilled medical coding
assistant trained to extract key lead terms
from clinical notes.

Your task is to analyze clinical notes and
exhaustively extract the most relevant lead
terms and their modifiers.

====== Now let's start! ======
Clinical Note: {{ note }}

Analyze the clinical note and extract the
most relevant lead terms.

Please reason step by step, and output your
final answer as a comma-separated list of
strings within <answer>...</answer>.
<think>
```

Index Navigator

```
You are a highly skilled medical coding
assistant specializing in structured code
extraction from clinical notes.

Your task is to analyze text for medical
terminology to output the few most relevant
terms from the alphabetical index.

====== Alphabetical Index ======
ID: {{ loop.index }} | Term: "{{ term }}" |
ID END: {{ loop.index }}

====== Now let's start! ======
Text: {{ query }}

Analyze the alphabetic index against the
text and select the IDs of the most relevant
 terms.

If no terms are relevant, output ID 0.

Please reason step by step, and output your
final answer as the IDs of the selected term
 within <answer>...</answer>.
<think>
```

Tabular Validator

```
You are a highly skilled medical coding
assistant specializing in structured code
extraction from clinical notes.

Your task is to analyze the clinical note
and identify the relevant candidate codes.

Note that a clinical note may record many
codes, but you are given a subset of
candidate codes, where none or one is
assignable.

====== Now let's start! ======
Clinical Note: {{ custom_tojson(note |
escape) }}

====== Guidelines ======
{{ guidelines }}

====== Candidate Codes ======
ID: {{ loop.index }} |  Code: {{ code }} |
ID END: {{ loop.index }}

Analyze the clinical note against the
guidelines to select the ID of the most
appropriate code.

If no relevant codes are found, output ID 0.

Please reason step by step, and output your
final answer as the ID of the selected code
within <answer>...</answer>.
<think>
```

Code Reconciler

```
You are a highly skilled medical coding
assistant specializing in structured code
extraction from clinical notes.

Note that a clinical note may record many
codes, but you are given a subset of
candidate codes where one or more are
assignable.

====== Now let's start! ======
Clinical Note: {{ note }}

====== Guidelines ======
{{ guidelines }}

====== Instructional Notes ======
{{ instructional_notes }}

====== Candidate Codes ======
ID: {{ loop.index }} |  Code: {{ code }} |
ID END: {{ loop.index }}

Analyze the clinical note against the
guidelines and the instructional notes to
select the IDs of all assignable codes,
prioritizing precision over recall.

Please reason step by step, and output your
final answer as the IDs of the selected
codes within <answer>...</answer>.
<think>
```

D Retrieval results

D.1 Dedicated Search Index

We employ a vector-based semantic search architecture using Qdrant[10] as the dedicated search index, selected to support sparse, dense, and hybrid search. The alphabetical terms are embedded and stored in a Qdrant index configured with Cosine distance and HNSW indexing (m=32, ef_construct=256). We use reciprocal rank fusion (RRF) scoring for hybrid search to combine cosine similarity results across models.

D.2 Retrieval results when using annotated evidence spans.

We simulate an upper bound for our retriever. We use the manually annotated evidence spans from the MDace dataset as search queries. Each span corresponds to codable text snippets identified by human coders and is directly matched against codes in the ICD hierarchy. This setup allows us to evaluate embedding-based models under the assumption of "optimal" codable concept extraction from clinical notes.

[10] https://qdrant.tech/documentation/

Model	P.	D.	@5	@10	@25	@50	@100
bm25	-	-	0.52	0.56	0.63	0.67	0.70
S-PubMedBert	110M	768	0.70	0.75	0.81	0.84	0.88
MedEmbed	335M	768	0.62	0.67	0.76	0.81	0.86
bm25+S-PubMedBert	110M	768	0.69	0.74	0.80	0.84	0.88

Table 4: Recall@k for different models on lead term search using cosine similarity. P. = Params, D. = Dim.

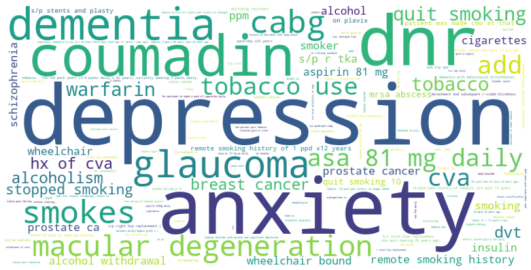

Figure 8: Word-cloud of all expert-annotated spans that the `Analyze` Agent failed to code in chapters Z00–Z99 (social and administrative factors) and F01–F99 (mental and behavioural disorders). Font size indicates corpus frequency.

D.3 Retrieval results when using 'Analyse' Agent.

We assess the 'Analyse' agent's ability to extract codeable information from clinical notes, using the text snippets it outputs as search queries. The agent processes the clinical note to identify codable information and outputs a list of phrases resembling indexable terms. These phrases serve as input queries to the semantic search index. By comparing recall@k against the results obtained with manually annotated evidence spans, we measure the agent's ability to approximate expert-level identification of codable content. This evaluation captures the compound challenge of accurate information extraction and effective retrieval under real-world constraints.

Model	P.	D.	@5	@10	@25	@50	@100
bm25	-	-	0.40	0.44	0.5	0.54	0.58
S-PubMedBert	110M	768	0.50	0.54	0.60	0.64	0.68
MedEmbed	335M	768	0.47	0.52	0.58	0.63	0.68
bm25+S-PubMedBert	110M	768	0.49	0.54	0.60	0.63	0.67

Table 5: Recall@k for different models on lead term search using cosine similarity. P. = Params, D. = Dim.

D.4 False negative evidence spans for Z00-Z99 and F01-F99

Figure 8 visualises the text evidence that the 'Analyze' Agent missed when analysing codeable text snippets for chapters Z00–Z99 and F01–F99.

E Fine-tuning results

E.1 Baseline results

We reproduce the PLM-ICD model following the implementation details provided by (Edin et al., 2024) and train it on MIMIC-IV to evaluate it on the test split of the MDace dataset. This serves as a baseline for comparison with our proposed method. The reproduction adheres to the original model's architecture and training setup to ensure comparability. PLM-ICD uses a BERT encoder pre-trained on PubMed to encode the text in chunks of 128 tokens, and these contextualized embeddings are fed to a cross-attention layer along with a learned representation for each code.

Table 6: Macro and micro F1 scores of the PLM-ICD model by document type on the MDace test split.

	F1 Score	
Document Type	**Macro**	**Micro**
Consult	0.09	0.20
ECG	0.29	0.56
Nursing	0.30	0.55
Nutrition	0.17	0.26
Case Management	0.17	0.30
Physician	0.23	0.41
Radiology	0.08	0.15
Rehab Services	0.12	0.19
General	0.21	0.35
Discharge Summary	0.27	0.55
Overall	**0.21**	**0.46**