CliniBench: A Clinical Outcome Prediction Benchmark for Generative and Encoder-Based Language Models

Paul Grundmann[1], Jan Frick[1], Dennis Fast[1], Thomas Steffek[1], Felix Gers[1], Wolfgang Nejdl[2], Alexander Löser[1],

[1]DATEXIS, Berlin University of Applied Sciences, Berlin, Germany
[2]Leibniz University Hannover, Hannover, Germany
Correspondence: jan.frick@bht-berlin.de

Abstract

With their growing capabilities, generative large language models (LLMs) are being increasingly investigated for complex medical tasks. However, their effectiveness in real-world clinical applications remains underexplored. To address this, we present CliniBench, the first benchmark that enables comparability of well-studied encoder-based classifiers and generative LLMs for discharge diagnosis prediction from admission notes in the MIMIC-IV dataset. Our extensive study compares 12 generative LLMs and 3 encoder-based classifiers and demonstrates that encoder-based classifiers consistently outperform generative models in diagnosis prediction. We assess several retrieval augmentation strategies for in-context learning from similar patients and find that they provide notable performance improvements for generative LLMs.

1 Introduction

The integration of transformer-based language models in clinical decision support systems is a rapidly advancing field, offering transformative potential for personalized healthcare, early diagnosis, and treatment planning (Laohawetwanit et al., 2024). Among these applications, language models are being studied for the task of diagnosis prediction from admission notes, an extreme multi-label classification task with unbalanced data (van Aken et al., 2021, 2022; Röhr et al., 2024; Grundmann et al., 2024; Figueroa et al., 2024; Gema et al., 2024). Solving this machine learning task requires resources and expertise, which are often scarce or do not exist in hospitals. Furthermore, privacy and data protection laws often require that models are trained and applied within the hospital, preventing data or model sharing.

Encoder-based classifiers (Encoder) currently represent the state-of-the-art for diagnosis prediction, with most research focusing on fine-tuning pretrained models. While they offer cost-efficient inference and strong task adaptation, their often fixed label space limits scalability, and they risk overfitting to training data. Rapid developments in medical research and the emergence of new diseases necessitate the continual adaptation of existing models. This is typically achieved via repeated fine-tuning, which in turn can lead to catastrophic forgetting. These challenges have fueled a growing interest in generative large language models (LLMs).

Generative LLMs offer key benefits such as zero-shot inference, dynamic label spaces, and in-context learning (ICL) (Brown et al., 2020). These features reduce the need for fine-tuning and result in a flexibility that encoder-based classifiers are missing. Therefore, generative LLMs should be particularly well suited for clinical applications, where labeled datasets are costly to create due to data protection laws. Despite their potential, generative models have not been systematically benchmarked against encoder-based classifiers for the diagnosis prediction task.

Learning at Test Time. While generative LLMs excel in adaptability and performance across diverse tasks, they are prone to hallucinations (i.e., factually incorrect or irrelevant outputs). To mitigate this limitation, they can be augmented with techniques like Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) and Chain-of-Thought (CoT) (Wei et al., 2022) prompting. RAG improves the reliability of generative models by linking them to external knowledge bases, improving the factual accuracy of their outputs. CoT prompting enhances logical consistency by encouraging step-by-step explanations, which helps with complex tasks. CoT requires only minimal adjustments to the model's prompt structure, while RAG leverages existing hospital databases. These strategies enable generative LLMs to become more robust and reliable,

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 5360–5378

making them a practical choice for healthcare applications, even in resource-constrained environments.

Contributions. In this paper, we introduce CliniBench, a comprehensive benchmark designed to simulate the unique challenges of clinical outcome prediction. CliniBench enables a systematic comparison of traditional state-of-the-art transformer encoders and emerging generative models. We summarize our contributions as follows:

- We publish the first benchmark[1] to compare encoders with generative LLMs for diagnosis prediction from admission notes based on MIMIC-IV (Johnson et al., 2023c).
- We evaluate the augmentation techniques RAG and CoT prompting and compare them with the performance of state-of-the-art fine-tuned encoder models.
- We provide an in-depth error analysis on a comprehensive set of experiments across various model configurations and augmentation strategies.

2 Related Work

Clinical outcome prediction is based solely on information available at hospital admission represents a distinct challenge compared to retrospective ICD coding or predefined text classification tasks. van Aken et al. (2021) introduce the task based on MIMIC-III (Goldberger et al., 2000; Johnson et al., 2016) to forecast discharge outcomes such as ICD codes, mortality, and length-of-stay. Naik et al. (2022) propose enhancing outcome prediction by retrieving relevant biomedical literature, grounding the predictions to improve both accuracy and interpretability. Röhr et al. (2024) extend this task to MIMIC-IV (Johnson et al., 2023c), adding new outcomes like patient routing and evaluating benchmark performance. Grundmann et al. (2024) examine data drift between MIMIC versions and highlight inconsistencies in ICD documentation practices, taxonomy, and label reliability. To improve label generalization and interpretability, van Aken et al. (2022) propose ProtoPatient, a prototypical network-based model. This approach was later extended by Figueroa et al. (2024) to better handle long-tail ICD distributions. Gema et al. (2024) demonstrate that fine-tuning LLMs for classification, rather than generation, can yield improved performance on admission-based tasks using parameter-efficient methods. Shoham and Rappoport (2024) focus on predicting readmission for three specific conditions, using LLMs trained on structured diagnosis descriptions from prior visits. In contrast to similar studies that utilize the full patient notes as inputs, the authors use textual descriptions of previously assigned diagnostic codes for their prediction.

2.1 Related Tasks

Two common well-studied, related tasks are ICD coding and medical question answering. Furthermore, as noted by Bedi et al. (2025), only five percent currently make use of real patient records and realistic clinical tasks.

ICD coding resembles an information extraction rather than a diagnosis prediction task because it incorporates all available information about a patient at discharge time. Related works include Mullenbach et al. (2018), Nguyen et al. (2023b), and Nguyen et al. (2023a), which focus on ICD coding of clinical notes from MIMIC-II, MIMIC-III and MIMIC-IV. Boyle et al. (2023) propose a tree-search method using the hierarchical structure of ICD codes, enabling superior performance on rare codes. Similarly, Ong et al. (2023) evaluate ChatGPT for ICD coding in retina clinics, achieving 70% accuracy but highlighting challenges such as hallucinations, underscoring the need for oversight in clinical applications.

Medical question answering. Beyond coding, LLMs are increasingly applied to medical question answering tasks. For instance, Ben Shoham and Rappoport (2024) and Xie et al. (2024) evaluate closed-source models on curated diagnosis-ranking benchmarks using small numbers of cases. Wu et al. (2024) propose using chain-of-thought prompting and span correction for clinical note error detection.

2.2 Learning at Inference Time

To mitigate some of the general limitations of LLMs, researchers have explored various techniques such as chain-of-thought prompting (Wei et al., 2022), few-shot learning (Parnami and Lee, 2022; Song et al., 2023), self-reflection (Asai et al., 2024; Jeong et al., 2024), self-verification (Weng et al., 2023; Gero et al., 2023) and guided decoding (Kuchnik et al., 2023; Scholak et al., 2021; Willard and Louf, 2023). For our evaluation, we focus on

[1] https://github.com/datexis/clinibench

RAG, CoT, and guided decoding to mitigate hallucinations.

Retrieval augmented generation enhances LLMs by incorporating external knowledge via in-context learning, improving accuracy, and reducing hallucinations. For example, Lu et al. (2024) introduced ClinicalRAG for reliable clinical decision support, Kresevic et al. (2024) used RAG to increase guideline interpretation accuracy to 99% and Unlu et al. (2024) applied RAG for efficient clinical trial screening, outperforming manual methods. While Xiong et al. (2024) benchmarked RAG systems, showing up to 18% accuracy improvements, Ngo et al. (2024) provided practical optimization guidelines for RAG systems. Finally, Li et al. (2025) show that chunking and fine-granular data selection for RAG improve performance on biomedical relation extraction and classification tasks.

Chain-of-Thought prompting enables step-by-step reasoning in LLMs, excelling in multi-step and symbolic tasks. Wei et al. (2022) introduce CoT and demonstrate its effectiveness in reasoning tasks. Zhang et al. (2023) automate CoT prompt generation to scale its application. Wang et al. (2023) tailor CoT for in-depth dialogue, enhancing coherence in responses. Sprague et al. (2024) analyze CoT's strengths in math and symbolic reasoning while highlighting its domain limitations.

Guided Decoding. To mitigate potential hallucinations that affect the parsability of model-generated output Kuchnik et al. (2023); Scholak et al. (2021) propose to limit the output token space of the LLM during generation given a finite state machine constructed from a regular expression and the respective context that has already been generated by the LLM. Other approaches, such as prompting the model to output a specific format or hierarchical decoding, do not guarantee that the model follows the respective format. Furthermore, approaches such as hierarchical decoding of the respective ICD codes directly (Boyle et al., 2023) do not make use of the textual generative properties of the LLM.

3 CliniBench

In order to compare encoder-based classifiers and generative LLMs we build the CliniBench benchmark depicted in Fig. 1. The benchmark consists of the task of predicting a patient's outcome in the form of discharge codes for a clinical note at admission.

3.1 MIMIC-IV Dataset

The source of medical texts is the Medical Information Mart for Intensive Care (MIMIC-IV) dataset. MIMIC-IV is grouped into two modules: HOSP and ICU. Data in the HOSP module are sourced from the hospital-wide electronic health record (EHR), while data in the ICU module are sourced from the clinical information system from the intensive care unit (ICU).

Admission Notes. To support clinical decision-making through diagnosis prediction, we limit the extracted information to relevant data known at admission time. We exclude sections containing outcome-related information obtained during the stay or after hospitalization, and exclude administrative or demographic fields that are de-identified or do not contribute to the diagnosis prediction task (see Section A.2 for complete section details).

ICD Discharge Codes. Both modules include detailed discharge codes derived from the International Classification of Diseases (ICD) coding system, documenting diseases and medical conditions (morbidity) associated with each admission. Depending on the date of admission, codes are available in either ICD-9 or ICD-10 formats (see Table 1). Given the hierarchical nature of the ICD system, we reduce the granularity of each code to three digits, allowing encoder-based models to learn from multiple examples rather than single instances during fine-tuning. This reduces the average number of codes per note to 13.98 while still providing adequate clinical information.

Dataset Construction. We derive CliniBench dataset from the MIMIC-IV-Note dataset (Johnson et al., 2023b) and ICD discharge codes from the MIMIC-IV dataset (Johnson et al., 2023a) following van Aken et al. (2021). Each instance corresponds to one hospital admission note (see Appendix A.2 for construction details).

Dataset Split. We generate a stratified split into training (70%), validation (10%), and test (20%) sets. The dataset comprises four partitions reflecting coding system and care setting: ICD-9/ICD-10 × HOSP/ICU. We exclude labels not present across all three splits from evaluation. We additionally report the notes-per-label cutoffs used to define frequency tertiles: Tail contains labels

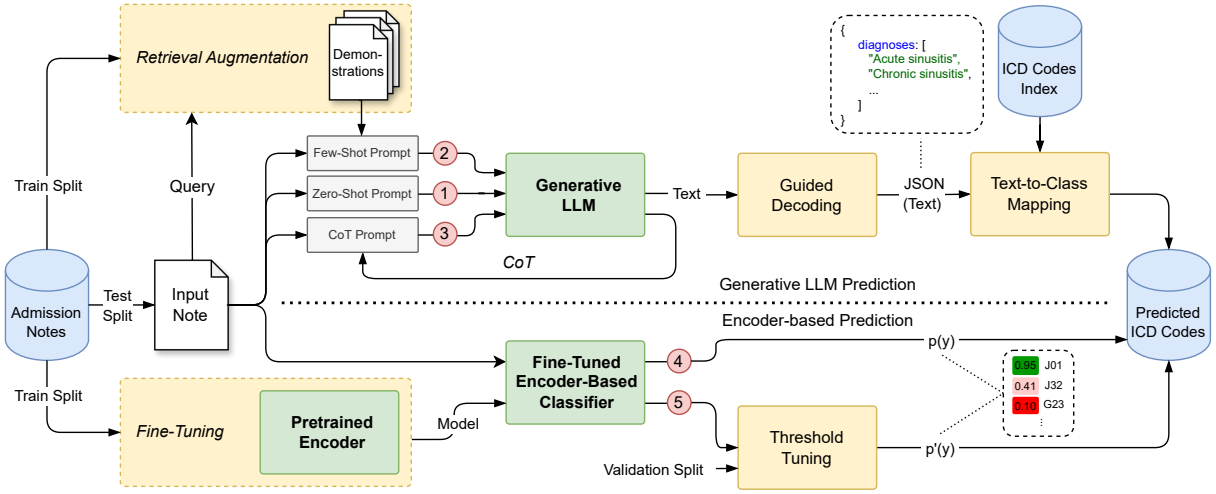

Figure 1: Architecture diagram of the CliniBench benchmark. Depicted are both model classes: Generative LLM (top) and encoder-based classifier (bottom). In addition to zero-shot evaluation (1), the LLM can be augmented via retrieval augmentation (2) and chain-of-thought (CoT) prompting (3) and predicts text (valid JSON) that is mapped to distinct labels. The encoder (bottom) predicts a probability distribution over the possible diagnoses (4) which can be threshold-tuned to improve performance (5).

	ICU		HOSP	
	ICD-9	ICD-10	ICD-9	ICD-10
#Train Adm. Notes	27007	18596	119025	66736
#Val Adm. Notes	3618	2399	15683	8912
#Test Adm. Notes	8089	5600	35930	20058
#Full Codes	6436	9610	8209	14362
#Short Codes	1038	1472	1105	1616
Avg. #Tokens/Adm. Note	670.4	624.9	681.5	687.5
Avg. #Codes/Adm. Note	14.6	18.4	10.0	12.9
#Adm. Notes/Code (Tail) $\leq$	8	7	17	10
#Adm. Notes/Code (Head) $>$	55	43	126	67

Table 1: Statistics of the four CliniBench partitions derived from MIMIC-IV admission notes. We report train/validation/test sizes, label set sizes, average label and token counts, and admission notes-per-label cutoffs for the frequency tertiles. Token counts are computed using the BioMedBERT tokenizer.

with frequency $\leq t_{tail}$, Head contains labels with frequency $> t_{head}$ and Body contains labels with $t_{tail} > freq \leq t_{head}$. Cutoffs are computed on the train split per partition. Detailed dataset statistics are provided in Table 1.

3.2 Model Classes and Augmentation

Fig. 1 illustrates the process of predicting ICD codes from admission notes using two primary approaches: A generative, LLM-based approach and an encoder-based approach.

Encoder. For the encoder-based prediction pipeline, we fine-tune a pretrained transformer to predict ICD diagnosis code probabilities. To ensure comparability with generative models, we evaluate using threshold-based metrics like recall and precision. Since thresholds can be noisy for rare classes, we tune them by maximizing the F_1 score on the Precision-Recall curve. We report details on the threshold-tuning in Section 4.1.

Generative LLMs. For the generative LLMs, we prompt each model to output a list of short diagnosis descriptions. We apply guided decoding (Willard and Louf, 2023) to ensure a machine-readable JSON format. These structured outputs are mapped to corresponding ICD codes using text-based matching. We investigate three distinct approaches: Zero-Shot, Few-Shot, and Zero-Shot with CoT prompting.

Zero-Shot. In the Zero-Shot setting, we input the admission notes with task-specific instructions. Thus, the model can only leverage knowledge acquired from its initial pretraining and the information available in the shown example.

Few-Shot. Generative LLMs can learn from few-shot examples provided in the prompt (Brown et al., 2020; Song et al., 2023; Parnami and Lee, 2022) without adapting the parameters (in-context-learning). For this adaptation, either the user or a system incorporates complementary knowledge in the prompt. This is especially useful for domain specific tasks that are not well represented in the pretraining data.

Chain-of-Thought. Finally, we apply chain-of-thought reasoning by prompting the model to pro-

duce reasoning steps prior to generating the list of diagnoses. We hypothesize that incorporating reasoning can improve the accuracy and diversity of predictions by mirroring the process of differential diagnosis in real-world clinical settings.

4 Experimental Setup

4.1 Encoder

Previous work has shown that encoders in the clinical setup benefit from in-domain pretraining (Lee et al., 2019). Therefore, we use two encoder models that are fine-tuned for clinical tasks: BioMedBert (Gu et al., 2021) and GatorTronS (Peng et al., 2023). Since preliminary results show a decline of model performance beyond their context length (see Fig. 5), we additionally evaluate a state-of-the-art long context encoder model, M2-BERT (Saad-Falcon et al., 2024).

Fine-tuning and hyperparameter optimization. We fine-tune the pretrained encoders on the train split and select the best performing using AUROC on the validation split using AdamW (Loshchilov and Hutter, 2019) as optimizer and a polynomial learning rate decay. For the hyperparameter optimization, we apply Bayesian optimization. For more details compare Table 4 and Table 7.

Threshold tuning. We optionally calibrate encoder outputs via class-wise threshold tuning on the validation split. For each label c, we evaluate F_1 over the sorted candidate threshold list T_c and choose $j = \arg\max F_c$; we set $\tau_c = (T_c[j-1] + T_c[j])/2$ (or $\tau_c = T_c[0]$ if $j = 0$). This midpoint rule avoids a fixed offset when thresholds are unevenly spaced. After calibration we still select the top-n labels by score; tuning only changes label ordering, primarily affecting low-frequency labels.

4.2 Generative LLMs

We evaluate open-weight generative LLMs that are capable of processing sequences exceeding 4096 tokens. This extended sequence capacity is crucial for evaluating models in the few-shot setting, as sequence length increases with each added demonstration. Consequently, we excluded both available medical LLMs MedAlpaca and Med42 from our analysis, as both are based on the Llama2 architecture, which limits their maximum sequence length to 2048 tokens. We selected the following models for evaluation: Llama 3.1 8B and 70B, Mistral 7B, Mistral Nemo (13B), Qwen2 7B and 72B. For each model, we evaluate both the instruction fine-tuned and the non-instruction-tuned variants, resulting in a total of 12 models. We explicitly do not benchmark closed-source commercial models due to constraints of the Physionet license [2].

Guided decoding. While generative models often produce valid JSON outputs and the correct number of prompted diagnoses, some of the evaluated models, especially those not fine-tuned with instruction data, frequently fail to do so. This limitation reduces overall task performance compared to encoder-based models. We report the performance of generating valid JSON output for all models in the Appendix in Section B.1. To mitigate this issue, we apply guided decoding (Willard and Louf, 2023) to enforce JSON output that contains exactly 20 strings. Each string represents a diagnosis description between 3 and 70 characters in length. To optimize performance, we limit the number of generated tokens to a maximum of 1500. Finally, we use greedy sampling with a temperature of 0.

Prompts. We use multiple prompts for each experiment: Zero-shot, zero-shot with CoT, and few-shot learning. The prompts consist each of a role description for the model and the respective task, further detailed in Section A.1.

Text-to-class mapping. Because LLMs often struggle to predict exact ICD codes (Boyle et al., 2023), we instead generate diagnostic descriptions and map them to ICD codes via their corresponding long or short descriptions. Based on a comparison of different approaches, we adopt NeuML-PubMedBERT[3] for text-to-label mapping. Details are reported in the Appendix in Section B.1. Finally, we truncate ICD codes to three digits to reduce label space and remove duplicates from the predictions.

4.3 Retrieval Models and Few-Shot Learning

We implement few-shot learning by incorporating one, three, or five demonstrations into the prompt. The effectiveness of in-context-learning depends on the selection of demonstrations (Liu et al., 2022). Thus, we use a retriever to find the most relevant candidates. The retriever takes one admission note

[2] https://physionet.org/content/mimiciv/view-license/2.2/

[3] https://huggingface.co/NeuML/pubmedbert-base-embeddings

as input query and returns the most similar admission notes from the training split. We conduct three different retrieval strategies and compare them to *random* and *gold* demonstrations. Namely, we use document-based semantic similarity, latent outcome similarity (classifier fine-tuned on diagnosis prediction task), and a term-based retrieving strategy.

Retriever selection and majority voting. To assess retrieval model performance for clinical outcome prediction, we evaluate them directly on the diagnosis prediction task. We apply majority voting over five retrieved training documents, selecting the top 20 most frequent labels, prioritizing documents with higher similarity in case of ties. We evaluate several semantic similarity retrievers based on S-BERT (Reimers and Gurevych, 2019) (see Section A.4 for complete list), selecting only the best performer for few-shot experiments to reduce computational costs. For latent outcome similarity retrieval, we use our fine-tuned BioMedBERT encoder (Section 4.1), hypothesizing that admission notes with similar outcomes cluster in embedding space. We also employ BM25 (Robertson and Zaragoza, 2009) for term-based retrieval.

Ideal retriever: *gold heuristic*. To assess the potential benefits of few-shot learning, we simulate an ideal retriever. Instead of retrieving based on semantic similarity, we sample demonstrations using the Otsuka similarity coefficient (Otsuka, 1936) of annotated ICD codes. Since this approach requires prior knowledge of the labels to identify similar patients, the experiment is designed solely to illustrate the upper limit of few-shot learning performance when leveraging similar patient examples.

Distractive demonstrations: *random*. Finally, we assess random sampling to determine the extent to which the LLM relies on the provided demonstrations and whether irrelevant examples negatively impact the performance of the generative LLM. This approach also enables us to evaluate whether the model effectively adapts to the structure of the presented demonstrations.

4.4 Evaluation Protocol and Metrics

Fixed-size top-n evaluation. We evaluate all methods in a fixed-size top-n setting, where each model outputs exactly n diagnosis codes per admission note. A fixed output size is required to ensure comparability between encoder-based classifiers, which assign a score to each label, and generative LLMs, which produce an explicit list of codes.

For encoder models, we rank labels by predicted score and select the top-n codes; when threshold tuning is applied, it affects score calibration and thus the ranking prior to top-n selection. For generative models, we constrain decoding to return a JSON list containing exactly n codes, thus simplifying guided decoding schema and reducing the maximum number of generated tokens per instance.

We choose n=20 as a single global output budget that yields stable comparisons across partitions and supports a recall-oriented operating point. This value is motivated by the observed label cardinality in CliniBench: across partitions, notes contain on average 10.0–18.4 short codes (Table 1), and our preliminary analysis indicates an F_1 plateau around this range when varying n. We therefore use n=20 to avoid systematic under-prediction on high-cardinality notes while keeping the output budget fixed across model classes. We treat n as a benchmark hyperparameter that can be adjusted for different precision–recall preferences.

Evaluation Metrics. We report macro-averaged recall, precision, and F_1 as main metrics. In addition, we report mean average precision (MAP), computed over the ranked list of predicted codes. Finally, we report main diagnosis accuracy (MD Acc.), i.e., whether the first annotated diagnosis code is contained in the predicted top-n list.

Upper ceiling metrics. A fixed top-n protocol induces metric ceilings: for notes with fewer than n codes, precision cannot reach 1.0; for notes with more than n codes, recall cannot reach 1.0. These ceilings depend only on the set size $|Y|$, with $\mathrm{Prec}_{\max}(Y) = \min(|Y|, n)/n$ and $\mathrm{Rec}_{\max}(Y) = \min(|Y|, n)/|Y|$. On the test split with n=20, the resulting dataset-level ceilings are Prec. $= 0.5566$ and Rec. $= 0.9824$.

5 Results

5.1 Zero-Shot

Encoder-based architectures consistently outperform their generative counterparts with the exception of the M2 BERT model.

Encoders excel especially for predicting the main diagnosis with a main diagnosis accuracy (MD Acc.) of over 80% on the HOSP split and 78.4% on the ICU split. When threshold tuning

Data	Model	Zero-Shot prompted					Zero-Shot prompted + CoT					Few-Shot 5 Demonstrations				
		Rec.	Prec.	MAP	MD Acc.	F1	Rec.	Prec.	MAP	MD Acc.	F1	Rec.	Prec.	MAP	MD Acc.	F1
HOSP	**Encoder-based Models**															
	BioMedBERT-110M	**30.74**	19.91	33.61	78.21	14.63			–					–		
	BioMedBERT-110M-tuned	22.91	29.55	19.34	59.16	21.74			–					–		
	GatorTronS-345M	30.45	22.70	**35.26**	**80.94**	18.46			–					–		
	GatorTronS-345M-tuned	28.61	**33.44**	22.06	65.22	**25.96**			–					–		
	M2-80M	14.90	08.51	24.34	60.83	4.54			–					–		
	M2-80M-tuned	11.72	12.05	14.80	40.47	11.16			–					–		
	Generative LLM															
	Llama 3.1-8B	16.61	12.16	12.88	35.22	11.06	14.85	10.89	11.49	37.96	09.68	21.76	16.00	21.33	47.23	12.41
	Llama 3.1-8B-Instruct	22.31	17.23	14.05	47.53	11.61	20.84	09.93	11.95	49.61	10.11	20.99	14.24	18.06	44.34	15.45
	Llama 3.1-70B	17.29	09.78	11.40	35.21	09.66	20.72	09.42	12.36	43.63	10.20	20.55	12.67	19.68	49.05	15.08
	Llama 3.1-70B-Instruct	23.04	10.40	12.48	52.20	11.32	23.70	10.00	12.53	51.39	10.59	21.89	14.43	18.02	45.49	15.82
	Mistral-7B	10.45	12.02	10.12	27.25	08.50	00.40	03.86	02.29	00.69	00.15	19.15	13.78	18.02	44.13	15.29
	Mistral-7B-Instruct	26.45	10.52	14.88	54.06	12.16	23.96	09.46	12.81	49.76	10.86	25.33	14.29	18.77	48.89	15.93
	Mistral-Nemo-13B	14.87	11.30	12.53	32.83	10.15	13.66	09.37	09.90	38.32	08.00	19.18	13.76	18.44	45.97	15.10
	Mistral-Nemo-13B-Instruct	22.23	09.55	12.34	50.35	10.38	21.19	09.22	10.44	51.61	09.57	21.15	15.28	19.09	48.41	16.05
	Qwen2-7B	22.86	11.32	15.12	43.37	12.26	21.04	11.62	14.92	42.01	12.04	17.70	14.94	18.81	43.40	15.31
	Qwen2-7B-Instruct	25.80	11.27	15.28	48.95	12.90	24.63	10.76	15.32	49.38	12.15	24.32	13.68	19.38	51.65	15.54
	Qwen2-72B	24.63	11.06	15.45	46.17	12.47	22.18	10.42	14.86	44.05	11.49	22.57	14.36	20.38	49.44	16.46
	Qwen2-72B-Instruct	27.24	11.09	17.28	51.58	13.13	25.66	10.91	15.28	50.24	12.75	25.29	15.82	21.51	53.00	17.57

Table 2: (**MAP**: *Mean Average Precision*, **MD Acc.**: *Main Diagnosis Accuracy*) Zero-shot performance on the HOSP split over the top-20 predictions. The reported metrics are macro averaged. We aggregate ICD-9 and ICD-10 and report an unweighted mean. CoT is not beneficial while instruct fine-tuning boosts performance. Threshold tuning decreases the performance of encoder models within the top-20 predictions. We depict the best performing models in bold and underline the best performing generative models in each category.

is applied, BioMedBERT and GatorTronS demonstrate strong performance decreases in recall, MD Acc., and MAP. This effect can be explained by the fact that we only measure the performance within the first 20 of an average of 47.18 predicted labels.

Generative LLMs benefit from instruction tuning, though they tend to lag behind encoder models. The addition of CoT reasoning yields inconsistent results for generative models, sometimes providing marginal improvements but often reducing MAP or recall. We hypothesize that CoT induced reasoning could distract the model from the actual task. This aligns with results from other work which shows that CoT is only beneficial for tasks that involve mathematical or symbolic reasoning (Sprague et al., 2024).

5.2 Retrieval Models

Table 5 in the appendix shows the performance of retrieval models using majority voting on the diagnosis prediction task. We observe that our fine-tuned classifier, BioMedBERT$_{ft}$, achieves the best performance. Notably, the baseline BM25 outperforms all retrieval models based on semantic similarity. Among the sentence-transformer models, performance differences are minimal, with PubMedBERT$_{NeuML}$ outperforming others. These findings suggest that using domain specific retrieval models provides a modest advantage for clinical tasks.

Data	Retriever	#FS	Rec.	Prec.	MAP	MD Acc.
HOSP	Zero-Shot prompted	0	**27.24**	11.09	17.28	51.58
	Random	1	21.53	11.77	15.14	42.66
		3	20.14	12.25	15.88	42.86
		5	18.90	12.32	17.75	41.59
	BM25	1	25.12	13.97	19.33	49.62
		3	25.48	15.17	21.53	52.32
		5	25.29	**15.83**	**22.02**	53.00
	PubmedBERT$_{NeuML}$	1	22.38	11.90	16.47	46.86
		3	22.20	13.35	17.63	50.97
		5	21.87	13.87	20.18	52.63
	BioMedBERT$_{ft}$	1	23.05	12.59	16.94	46.59
		3	23.52	13.89	20.35	51.24
		5	24.25	14.63	20.89	**53.04**
	Gold Heuristic	1	27.04	15.80	20.97	51.36
		3	30.43	19.49	28.11	60.04
		5	32.04	20.77	31.90	62.62

Table 3: Few-shot performance of Qwen2-72B-Instruct averaged over ICD-9 and ICD-10 splits using the depicted retrieval models. Metrics are macro averaged. BioMedBERT$_{ft}$ is a custom fine-tuned encoder model on the respective training split of the dataset. (**#FS**: number of few-shot demonstrations, **MD Acc.**: Main diagnosis accuracy)

5.3 Few-Shot

As shown in Table 2, Table 3 and Fig. 4 (ICU scores reported in Appendix Table 11 and Table 10), providing generative LLMs with even a single example enhances their performance compared to zero-shot settings. This improvement is likely because the demonstration helps align the model more closely with the task, particularly benefiting non-instruct variants of generative models. Adding more demonstrations continues to boost performance, albeit with diminishing gains.

Impact of retrieval quality. The *gold* heuristic provides an idealized upper boundary for performance by retrieving the most relevant samples (see Table 5). It achieves the highest metrics across both splits, with the main diagnosis accuracy reaching 62.62% for HOSP and 61.66% for ICU, along with substantial MAP improvements. Among the retrievers, BM25 emerges as a robust baseline, consistently delivering competitive MAP and main diagnosis accuracy, outperforming NeuML-PubMedBERT and BioMedBERT$_{ft}$ in several configurations. Task-specific fine-tuning provides a slight advantage for BioMedBERT$_{ft}$, which outperforms NeuML-PubMedBERT. Random retrieval degrades the performance of generative models, underscoring the importance of relevance-driven retrieval strategies for downstream tasks.

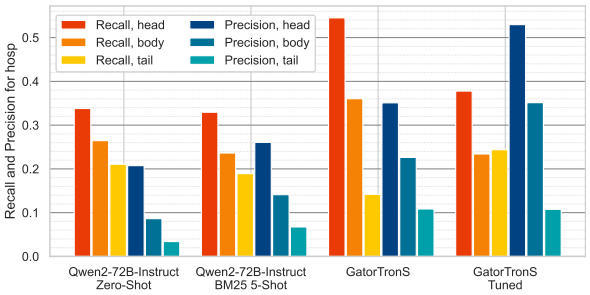

Figure 2: Macro recall and precision by model and classes grouped into tertiles by class frequency aggregated over both HOSP splits (ICD-9 and ICD-10).

Performance across different label frequencies. The results in Fig. 2 (see also Appendix Fig. 6) highlight variations in performance across labels with different frequencies. We observe that general performance is better for frequent labels. We find that retrieval augmentation boosts precision and reduces recall for all tertiles, particularly for less frequent labels. Generative models achieve a higher recall for rare codes than encoder-based classifiers. Finally, we observe that threshold tuning GatorTronS improves recall in the tail, while decreasing it in the head tertile.

Impact of model size. We observe that the number of parameters does not have a strong impact on performance. Further, we find that few-shot learning especially benefits smaller models, increasing the performance drastically in contrast to larger models. We hypothesize that guided decoding reduces the necessity of capacity in the larger models, making small models suitable for the task. Furthermore, we find that instruct fine-tuning has a larger effect on performance compared to the number of parameters (see Fig. 9), narrowing the gap between small and large models.

5.4 Error Analysis

In addition to presenting quantitative results, we conduct a qualitative analysis of 20 randomly sampled examples with especially low macro recall from Qwen2-72B and Qwen2-72B-Instruct in zero-shot, few-shot (BM-25), and chain-of-thought settings. From our analysis, we deduce the following error classes:

Low variance outputs in generative LLMs. We observe that generative models sometimes generate text with little to no variance (e.g., *Lung Cancer, Lung Carcinoma, Cancer*), leading to fewer predicted codes after deduplication. This behavior appears mainly in few-shot augmentation (80% of one-shot, 75% of five-shot predictions) and 50% of CoT predictions. As shown in Fig. 3, all models predict fewer than the expected 20 codes on average, suggesting this issue exists across scenarios. Non-instruct fine-tuned models generate more varied output and thus more diverse diagnoses.

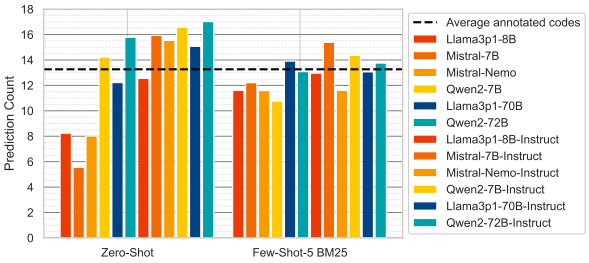

Figure 3: Average number of predicted codes after deduplication for zero-shot and few-shot experiments. Small LLMs are represented in shades of red, large ones in shades of blue, with instruction-tuned models highlighted in a shaded pattern.

Unrelated generated text. We find that LLMs frequently produce valid but irrelevant text (e.g., procedures instead of diagnoses) or non-meaningful text (e.g., *diagnosis 1, diagnosis 2, diagnosis 3*). In zero-shot settings with non-instruct models, 65% of errors originate from unrelated meaningful output, where models often output procedures or copy parts of the admission note. Interestingly, a single few-shot example increases non-meaningful text generation in 20% of cases, suggesting the model focuses on structural rather than task-specific elements.

6 Future Work

To address the limitations identified in our study, we propose several directions for future research that could substantially improve the performance of generative models for clinical outcome prediction.

Output Verification and Sampling To address low-variance and unrelated outputs, we suggest adding LLM-based verification mechanisms (Gero et al., 2023) to filter predictions and reduce duplicates. Additionally, probabilistic sampling strategies (top-p, top-k, beam search) could increase output diversity.

Enhanced Retrieval Strategies Given that latent outcome similarity outperforms semantic similarity and the gold heuristic surpasses both, future work should augment retrieval with additional clinical information beyond admission notes, such as laboratory results or imaging data, to improve demonstration selection for few-shot learning.

Domain Adaptation and Continual Learning To enable effective chain-of-thought reasoning, we suggest adapting models using clinical guidelines and datasets that capture clinicians' differential diagnosis processes (Wang et al., 2025). Future research should also explore efficient fine-tuning strategies that balance performance with computational costs for scenarios requiring frequent model updates in response to evolving medical knowledge.

7 Conclusion

This study introduces CliniBench, a benchmark designed to comprehensively evaluate and compare generative LLMs and encoder-based classifiers for diagnosis prediction. Our findings underscore that the encoders maintain superior performance. However, generative LLMs exhibit promising adaptability through in-context learning strategies which are advantageous in resource-constrained or rapidly evolving environments such as the medical domain. CliniBench provides a systematic framework to analyze the limitations of generative models, particularly high inference costs, lack of output diversity, and susceptibility to produce unrelated output. By highlighting necessary improvements in retrieval strategies, sampling techniques, and domain-specific model adaptations, CliniBench paves the way for future research to bridge the performance gap and unlock the full potential of generative LLMs in clinical decision support systems.

8 Acknowledgements

We thank the reviewers and the Area Chair for their constructive feedback, which improved this work. This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) ProjectID 528483508- FIP 12, as well as the European Union under the grant project 101079894 (COMFORT- Improving Urologic Cancer Care with Artificial Intelligence Solutions), as well as by BMWe SOOFI, Grant Agreement 13IPC040D. The views expressed are solely those of the authors and do not necessarily reflect those of the European Union or European Health and Digital Executive Agency (HaDEA); neither is responsible for them.

9 Limitations

Our findings should be interpreted within the context of several study-specific limitations. First, our experiments rely on a single large-scale clinical dataset of admission notes, which may not represent the diversity of healthcare systems, patient populations, or note-taking styles. Second, the evaluation focuses on diagnosis prediction from admission-time text and does not assess other clinically relevant outcomes (e.g., treatment recommendations or prognostic reasoning). Third, admission notes frequently exceed the context length of standard encoder models; for encoders with limited attention window, we truncate inputs to the first 512 tokens, which may omit clinically relevant information and bias performance toward evidence appearing early in the document. We partially mitigate this by additionally evaluating a long-context encoder (M2-BERT), but a systematic comparison of long-document strategies (e.g., chunking or hierarchical encoding) is outside the scope of this benchmark. Furthermore, we restrict our evaluation to open-weight models that can be executed locally. Due to the PhysioNet credentialed data use agreement for MIMIC-IV, we do not send protected note text to third-party commercial APIs and therefore do not benchmark closed-source models. As a result, our comparisons do not cover proprietary systems. Finally, while we evaluate multiple architectures and prompting strategies, our hyperparameter choices and model selection may not reflect optimal configurations for every method. Future work should explore broader tuning bud-

gets and additional modeling choices to assess the robustness and generalizability of our conclusions.

10 Ethical Considerations

The use of language models for diagnosis prediction raises important ethical concerns. For example, inaccurate or biased predictions can negatively impact patient care, as admission notes may lack critical information. It is important to note that an automatic diagnosis prediction system should never be seen as a replacement for medical personnel. Rather, it should be used as a supporting tool in the differential diagnosis process. Another major ethical concern is data privacy, as third-party data access can lead to serious consequences, including financial harm, discrimination, and compromised care. Finally, there is a risk that ICD code prediction might be misused to maximize profits, prioritizing financial incentives over patient care. Addressing these concerns is vital for the safe and effective use of AI in healthcare in the future.

References

Miriam Anschütz, Diego Miguel Lozano, and Georg Groh. 2023. This is not correct! negation-aware evaluation of language generation systems. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 163–175, Prague, Czechia. Association for Computational Linguistics.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Suhana Bedi, Yutong Liu, Lucy Orr-Ewing, Dev Dash, Sanmi Koyejo, Alison Callahan, Jason A. Fries, Michael Wornow, Akshay Swaminathan, Lisa Soleymani Lehmann, Hyo Jung Hong, Mehr Kashyap, Akash R. Chaurasia, Nirav R. Shah, Karandeep Singh, Troy Tazbaz, Arnold Milstein, Michael A. Pfeffer, and Nigam H. Shah. 2025. Testing and evaluation of health care applications of large language models: A systematic review. *JAMA*, 333(4):319–328.

Ofir Ben Shoham and Nadav Rappoport. 2024. Cpllm: Clinical prediction with large language models. *PLOS Digital Health*, 3(12):1–15.

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. *arXiv preprint*. ArXiv:2310.06552 [cs].

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Pritam Deka, Anna Jurek-Loughrey, and P Deepak. 2022. Improved methods to aid unsupervised evidence-based fact checking for online health news. *Journal of Data Intelligence*, 3(4):474–504.

Alexei Figueroa, Jens-Michalis Papaioannou, Conor Fallon, Alexandra Bekiaridou, Keno Bressem, Stavros Zanos, Felix Gers, Wolfgang Nejdl, and Alexander Löser. 2024. Boosting long-tail data classification with sparse prototypical networks. In *Machine Learning and Knowledge Discovery in Databases. Research Track*, pages 434–449, Cham. Springer Nature Switzerland.

Aryo Pradipta Gema, Pasquale Minervini, Luke Daines, Tom Hope, and Beatrice Alex. 2024. Parameter-Efficient Fine-Tuning of LLaMA for the Clinical Domain. *arXiv preprint*. ArXiv:2307.03042 [cs].

Zelalem Gero, Chandan Singh, Hao Cheng, Tristan Naumann, Michel Galley, Jianfeng Gao, and Hoifung Poon. 2023. Self-Verification Improves Few-Shot Clinical Information Extraction. *arXiv preprint*. ArXiv:2306.00024 [cs].

A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley. 2000. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):E215–220.

Paul Grundmann, Jens-Michalis Papaioannou, Tom Oberhauser, Thomas Steffek, Amy Siu, Wolfgang Nejdl, and Alexander Loeser. 2024. Data drift in clinical outcome prediction from admission notes. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4381–4391, Torino, Italia. ELRA and ICCL.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. *ACM Trans. Comput. Healthcare*, 3(1):2:1–2:23.

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, Maximilian Werk, Nan Wang, and Han Xiao. 2023. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents. *Preprint*, arXiv:2310.19923.

Minbyul Jeong, Jiwoong Sohn, Mujeen Sung, and Jaewoo Kang. 2024. Improving medical reasoning

through retrieval and self-reflection with retrieval-augmented large language models. *Bioinformatics (Oxford, England)*, 40(Suppl 1):i119–i129.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. 2023a. MIMIC-IV (version 2.2).

Alistair Johnson, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. 2023b. MIMIC-IV-Note: Deidentified free-text clinical notes (version 2.2).

Alistair E. W. Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J. Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, Li-wei H. Lehman, Leo A. Celi, and Roger G. Mark. 2023c. 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, 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):160035.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *CoRR*, abs/2001.08361.

Simone Kresevic, Mauro Giuffrè, Milos Ajcevic, Agostino Accardo, Lory S. Crocè, and Dennis L. Shung. 2024. Optimization of hepatological clinical guidelines interpretation by large language models: a retrieval augmented generation-based framework. *npj Digital Medicine*, 7(1):1–9. Publisher: Nature Publishing Group.

Michael Kuchnik, Virginia Smith, and George Amvrosiadis. 2023. Validating large language models with relm. In *Proceedings of the Sixth Conference on Machine Learning and Systems, MLSys 2023, Miami, FL, USA, June 4-8, 2023*. mlsys.org.

Thiyaphat Laohawetwanit, Daniel Pinto, and Andrey Bychkov. 2024. A survey analysis of the adoption of large language models among pathologists. *American journal of clinical pathology*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Mingchen Li, Halil Kilicoglu, Hua Xu, and Rui Zhang. 2025. Biomedrag: A retrieval augmented large language model for biomedicine. *J. Biomed. Informatics*, 162:104769.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yuxing Lu, Xukai Zhao, and Jinzhuo Wang. 2024. ClinicalRAG: Enhancing clinical decision support through heterogeneous knowledge retrieval. In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 64–68, Bangkok, Thailand. 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.

Aakanksha Naik, Sravanthi Parasa, Sergey Feldman, Lucy Lu Wang, and Tom Hope. 2022. Literature-augmented clinical outcome prediction. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 438–453. Association for Computational Linguistics.

Nghia Trung Ngo, Chien Van Nguyen, Franck Dernoncourt, and Thien Huu Nguyen. 2024. Comprehensive and practical evaluation of retrieval-augmented generation systems for medical question answering. *Preprint*, arXiv:2411.09213.

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

Thanh-Tung Nguyen, Viktor Schlegel, Abhinav Ramesh Kashyap, and Stefan Winkler. 2023b. A

two-stage decoder for efficient ICD coding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4658–4665, Toronto, Canada. Association for Computational Linguistics.

Joshua Ong, Nikita Kedia, Sanjana Harihar, Sharat Chandra Vupparaboina, Sumit Randhir Singh, Ramesh Venkatesh, Kiran Vupparaboina, Sandeep Chandra Bollepalli, and Jay Chhablani. 2023. Applying large language model artificial intelligence for retina International Classification of Diseases (ICD) coding. *Journal of Medical Artificial Intelligence*, 6(0). Number: 0 Publisher: AME Publishing Company.

Yanosuke Otsuka. 1936. The faunal character of the japanese pleistocene marine mollusca, as evidence of the climate having become colder during the pleistocene in japan. *Bulletin of the Biogeographical Society of Japan (in Japanese)*, pages 165–170.

Archit Parnami and Minwoo Lee. 2022. Learning from few examples: A summary of approaches to few-shot learning. *ArXiv*, abs/2203.04291.

Cheng Peng, Xi Yang, Aokun Chen, Kaleb E Smith, Nima PourNejatian, Anthony B Costa, Cheryl Martin, Mona G Flores, Ying Zhang, Tanja Magoc, Gloria Lipori, Duane A Mitchell, Naykky S Ospina, Mustafa M Ahmed, William R Hogan, Elizabeth A Shenkman, Yi Guo, Jiang Bian, and Yonghui Wu. 2023. A study of generative large language model for medical research and healthcare.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Preprint*, arXiv:1908.10084.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3:333–389.

Tom Röhr, Alexei Figueroa, Jens-Michalis Papaioannou, Conor Fallon, Keno Bressem, Wolfgang Nejdl, and Alexander Löser. 2024. Revisiting clinical outcome prediction for MIMIC-IV. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, pages 208–217, Mexico City, Mexico. Association for Computational Linguistics.

Jon Saad-Falcon, Daniel Y. Fu, Simran Arora, Neel Guha, and Christopher Ré. 2024. Benchmarking and Building Long-Context Retrieval Models with LoCo and M2-BERT. *arXiv preprint*. ArXiv:2402.07440 [cs].

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021. PICARD: parsing incrementally for constrained auto-regressive decoding from language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 9895–9901. Association for Computational Linguistics.

Ofir Ben Shoham and Nadav Rappoport. 2024. CPLLM: Clinical Prediction with Large Language Models. *arXiv preprint*. ArXiv:2309.11295 [cs].

Yisheng Song, Ting Wang, Puyu Cai, Subrota K. Mondal, and Jyoti Prakash Sahoo. 2023. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *ACM Comput. Surv.*, 55(13s).

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2024. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. *Preprint*, arXiv:2409.12183.

Ozan Unlu, Jiyeon Shin, Charlotte J. Mailly, Michael F. Oates, Michela R. Tucci, Matthew Varugheese, Kavishwar Wagholikar, Fei Wang, Benjamin M. Scirica, Alexander J. Blood, and Samuel J. Aronson. 2024. Retrieval-augmented generation–enabled gpt-4 for clinical trial screening. *NEJM AI*, 1(7):AIoa2400181.

Betty van Aken, Jens-Michalis Papaioannou, Manuel Mayrdorfer, Klemens Budde, Felix Gers, and Alexander Loeser. 2021. Clinical outcome prediction from admission notes using self-supervised knowledge integration. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 881–893, Online. Association for Computational Linguistics.

Betty van Aken, Jens-Michalis Papaioannou, Marcel Naik, Georgios Eleftheriadis, Wolfgang Nejdl, Felix Gers, and Alexander Loeser. 2022. This patient looks like that patient: Prototypical networks for interpretable diagnosis prediction from clinical text. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 172–184, Online only. Association for Computational Linguistics.

Bowen Wang, Jiuyang Chang, Yiming Qian, Guoxin Chen, Junhao Chen, Zhouqiang Jiang, Jiahao Zhang, Yuta Nakashima, and Hajime Nagahara. 2025. Direct: Diagnostic reasoning for clinical notes via large language models. *Preprint*, arXiv:2408.01933.

Hongru Wang, Rui Wang, Fei Mi, Yang Deng, Zezhong Wang, Bin Liang, Ruifeng Xu, and Kam-Fai Wong. 2023. Cue-cot: Chain-of-thought prompting for responding to in-depth dialogue questions with LLMs. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. Large language models are better reasoners

with self-verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2550–2575, Singapore. Association for Computational Linguistics.

Brandon T. Willard and Rémi Louf. 2023. Efficient guided generation for large language models. *CoRR*, abs/2307.09702.

Zhaolong Wu, Abul Hasan, Jinge Wu, Yunsoo Kim, Jason Cheung, Teng Zhang, and Honghan Wu. 2024. KnowLab_AIMed at MEDIQA-CORR 2024: Chain-of-though (CoT) prompting strategies for medical error detection and correction. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, pages 353–359, Mexico City, Mexico. Association for Computational Linguistics.

Qianqian Xie, Qingyu Chen, Aokun Chen, Cheng Peng, Yan Hu, Fongci Lin, Xueqing Peng, Jimin Huang, Jeffrey Zhang, Vipina Keloth, Xinyu Zhou, Lingfei Qian, Huan He, Dennis Shung, Lucila Ohno-Machado, Yonghui Wu, Hua Xu, and Jiang Bian. 2024. Me llama: Foundation large language models for medical applications. *Preprint*, arXiv:2402.12749.

Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. 2024. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6233–6251, Bangkok, Thailand. Association for Computational Linguistics.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.

A Appendix

A.1 Prompts

For few-shot demonstrations, we provide the admission notes together with their annotated diagnosis short descriptions in the desired JSON format. We use the following prompt structure for each respective task setting:

Zero-shot: "You are a medical professional. Given an admission note for a patient, present a list of possible diagnoses for the patient. The admission note is as follows: {note}."

Zero-shot with CoT: "You are a medical professional. Given an admission note for a patient, present a list of possible diagnoses for the patient. The admission note is as follows: {note}. TASK: Solve this task step by step and give an explanation in maximum one or two sentences for each diagnosis decision."

Few-shot: "You are a medical professional. Given an admission note for a patient, present a list of possible diagnoses for the patient. Similar patients look like this: {few_shots.json()}. The admission note is as follows: {note}. Give the diagnoses following the schema from the examples."

We hypothesize that providing the model with comprehensive information about the ICD ontology would enhance its performance. However, the full set of code descriptions exceeds the maximum context length of all evaluated models. Using the Qwen2.5 tokenizer, the descriptions of all possible ICD-9 codes require 167,679 tokens, and those of ICD-10 codes require 1,369,475 tokens. Consequently, we restrict the input to the task prompt, admission note and few-shot examples.

CoT implementation. For chain-of-thought reasoning, we modify the prompt to require the model to articulate specific reasoning steps before generating the list of diagnoses descriptions. Additionally, we adapt the expected JSON schema to include a reasoning field preceding the list of diagnoses. This ensures that the model first produces reasoning steps, followed by the diagnosis descriptions.

A.2 Construction of the Admission Notes

To avoid data leakage and maintain temporal consistency, we carefully select which sections of clinical notes to include based on whether the information would realistically be available at admission time.

Included sections with information available at admission time: *Chief Complaint, Major Surgical or Invasive Procedure, Allergies, History of Present Illness, Past Medical History, Social History, Family History, Physical Exam at Admission, and Medication at Admission*

Discarded sections that contain information not available at admission time: *Physical Exam during Stay and at Discharge, Pertinent Results, Brief Hospital Course, Medication at Discharge, Discharge Disposition, Facility, Discharge Diagnosis, Discharge Condition, Discharge Instructions, and Followup Instructions*

Sections that do not contribute to the task or prediction and are therefore discarded: *Name, Unit No, Admission Date, Discharge Date, Date of Birth, Sex, Service, and Attending. This systematic approach ensures that our models are evaluated under realistic clinical conditions where only admission-time information is available for out-*

come prediction

A.3 HPO and Threshold Tuning

See Table 4 for a breakdown of our hyperparameter optimization.

A.4 Retrieval Model Details

The retrieval evaluation enables us to gauge the impact of adding a generative LLM on top of the retrieved few-shot results. We evaluate the following semantic similarity retrievers based on the S-BERT architecture (Reimers and Gurevych, 2019): NeuML-PubmedBERT, jina-base-v2 (Günther et al., 2023), all-mpnet-base-v2[4], all-mpnet-base-v2-neg (Anschütz et al., 2023), BiomedBERT (Gu et al., 2021), and S-PubMedBert-MS-MARCO (Deka et al., 2022). For the latent outcome similarity retrieval, we use the CLS token from the last embedding layer of our fine-tuned baseline encoder (BioMedBERT) as a representation of the admission note.

A.5 Computational Cost Analysis

The computational costs differ dramatically between model types. Training a BERT-base model on the ICD-10 Hosp dataset requires about 9.51 PFLOPs per epoch (Kaplan et al., 2020), or 28.53 PFLOPs when including the backward pass. With a maximum of 20 training epochs, this totals 570.6 PFLOPs for our experiments. By contrast, simply evaluating the ICD-10 Hosp test set with generative models is far more expensive: Qwen2-72B requires approximately 4210.96 PFLOPs, while Llama 3.1 8B needs 252.1 PFLOPs. The smaller BERT-based model only uses only 2.89 PFLOPs for inference on the same test set (see Table 6). This demonstrates that generative models require orders of magnitude more computation than encoder-based models, even for inference on a small test set. These estimates ignore several real-world considerations, such as infrastructure requirements and hardware utilization. Larger generative models demand substantially more hardware during inference, which might remain underutilized between tasks, though these costs could be partially offset by leveraging techniques such as pruning and quantization.

[4] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

Parameter			Learning Rate	Warmup Steps	Weight Decay	Decay Steps	Batch Size	Mean Pooling
Ranges			$[1e{-}6,\ 1e{-}3]$	$[0, 5000]$	$[1e{-}6,\ 1e{-}3]$	$[1, 2, ..7] * 10000$	$[8, 64]$	$[False, True]$
BioMedBERT	HOSP	ICD-9	0.000131	994	0.026853	50000	32	False
		ICD-10	0.000526	146	0.095573	10000	64	False
	ICU	ICD-9	0.000305	28	0.020785	10000	64	False
		ICD-10	0.000158	374	0.008371	50000	16	False
GatorTronS	HOSP	ICD-9	0.000086	2185	0.034802	50000	16	False
		ICD-10	0.000141	665	0.002609	40000	16	False
	ICU	ICD-9	0.000098	4854	0.039681	50000	16	False
		ICD-10	0.000091	4691	0.001923	40000	16	False
M2-BERT	HOSP	ICD-9	0.000119	1728	0.000390	30000	8	False
		ICD-10	0.000029	1698	0.000003	70000	8	False
	ICU	ICD-9	0.000121	2376	0.000001	50000	16	False
		ICD-10	0.000244	3620	0.000001	60000	16	True

Table 4: Hyperparameter ranges and best found hyperparameters for the evaluated encoder models. We perform 20 HPO runs using Bayesian optimization for each combination of model and dataset.

Data	Model	Rec.	Prec.	MAP	MD Acc.
HOSP	BM25	14.73	8.73	16.97	45.22
	BioMedBERT$_{ft}$	**18.36**	**9.97**	**19.36**	**49.16**
	BioMedBERT	07.07	03.98	12.52	37.43
	all-mpnet-base-v2	12.86	07.58	14.74	42.55
	all-mpnet-base-v2-neg	10.75	06.59	13.82	40.59
	jina-base-v2-512	13.42	08.05	15.33	43.61
	jina-base-v2-2048	12.69	08.16	15.51	43.80
	PubMedBERT$_{NeuML}$	13.57	08.13	15.89	44.25
	S-PubMedBert$_{MS-MARCO}$	11.64	07.40	14.84	42.82
	M2-BERT	02.67	01.45	08.15	26.23
	GatorTronS	02.87	01.38	09.20	21.27
	Random	01.96	01.06	07.04	23.66
	Gold Heuristic	31.67	23.45	40.93	73.36
ICU	BM25	09.57	09.11	18.39	42.25
	BioMedBERT$_{ft}$	**11.73**	**09.92**	**19.99**	**45.08**
	BioMedBERT	05.11	04.63	15.27	36.88
	all-mpnet-base-v2	08.45	07.94	17.47	42.10
	all-mpnet-base-v2-neg	07.16	07.14	16.47	39.29
	jina-base-v2-512	08.38	08.23	17.89	41.96
	jina-base-v2-2048	08.04	07.99	17.91	42.18
	PubMedBERT$_{NeuML}$	09.12	08.40	18.61	42.69
	S-PubMedBert$_{MS-MARCO}$	07.82	07.65	17.56	41.24
	M2-BERT	02.93	02.20	11.71	28.03
	GatorTronS	03.34	02.73	12.81	25.54
	Random	02.47	01.90	10.49	26.14
	Gold Heuristic	19.98	19.85	37.69	64.41

Table 5: Independent analysis of retrieval models on the diagnosis prediction task. Metrics are macro averaged and the results are averaged over the respective ICD-9 and ICD-10 splits. A fine-tuned encoder works best (bold), followed by BM25 (underlined). The domain adapted PubMedBERT$_{NeuML}$ performs best from the sentence transformer models.

		ICU		HOSP	
Model	#Parameters	ICD 9	ICD 10	ICD 9	ICD 10
BioMedBERT	110M	1.17	0.73	5.23	2.89
GatorTronS	345M	3.45	2.16	15.41	8.53
M2 BERT	80M	1.12	0.72	4.97	2.80
Llama 3.1 8B	8B	103.08	64.68	454.58	252.1
Llama 3.1 70B	70B	901.93	565.97	3977.57	2206.30
Mistral 7B	7B	109.79	68.84	484.08	267.82
Mistral Nemo 13B	13B	181.99	113.66	802.34	441.28
Qwen 2 7B	7B	93.15	58.27	409.40	226.76
Qwen 2 72B	72B	599.37	958.10	2332.40	4210.96

Table 6: Inference cost in PFLOPs for each test split using the respective tokenizer of the model. We approximate the number of FLOPs based on (Kaplan et al., 2020) by multiplying the number of tokens of each dataset with the number of parameters of the model multiplied by 2. Qwen2, due to its more efficient tokenizer, reduces compute costs compared to Llama 3.1 by almost 40%.

B Additional Results

Model	Data		AUROC
BioMedBERT	HOSP	ICD-9	90.12
		ICD-10	88.01
	ICU	ICD-9	83.59
		ICD-10	80.38
GatorTronS	HOSP	ICD-9	91.96
		ICD-10	89.43
	ICU	ICD-9	86.23
		ICD-10	82.17
M2-BERT	HOSP	ICD-9	85.82
		ICD-10	81.05
	ICU	ICD-9	77.99
		ICD-10	75.08

Table 7: AUROC performance of the fine-tuned encoder models.

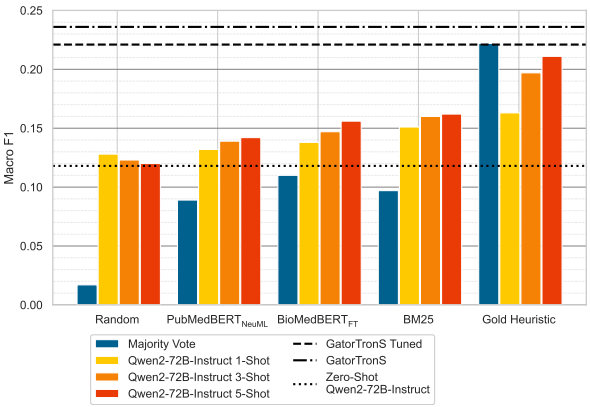

Figure 4: Macro F_1 (averaged over all datasets) of majority voting, best encoder, and best generative LLM with various amounts of demonstrations. Categories on the x-axis show different retrieving strategies for few-shot experiments. Horizontal lines depict the performance of the best zero-shot generative model and the best encoder models.

B.1 Text-to-class Mapper Performance.

We find that generative models often output diagnostic descriptions that exactly match the short or long descriptions of the respective ICD codes. This allows us to independently compare the performance of exact matched descriptions and codes that are mapped via the class mapper shown in Section B.1. For Qwen2-72B Instruct in the zero-shot scenario, only 5% of generated descriptions match ICD codes exactly; with 1, 3, and 5 few shots, this number rises to 32%, 51%, and 63% respectively. For the non-instruct fine-tuned version of the model, the number of exact matches rises even further to 56%, 70% and 76%. This indicates an adaption to the task through the provided few-shot examples. Furthermore, it shows that the model has seen the respective ICD short or long descriptions during pre-training and can correctly replicate them when it is provided with enough context. Among 5-shot predictions of Qwen2-72B Instruct, directly matched codes are correct 41% of the time, while mapped ones reach 20%. For Qwen2-7B, mapped codes retain 94% of the accuracy of directly matched ones (30% vs. 32%). While the accuracy of mapped codes is lower, their generated text contains enough relevant information that enables the class-mapper to resolve the correct code. This gap in performance is closely tied to the quality of the generated text. Accurate mappings result from meaningful clinical descriptions (e.g., *Hypertensive heart disease*, *Hypotension*), while nonsensical outputs (e.g., *diagnosis 1*, *diagnosis 2*, ___ etc.) almost never yield correct codes.

Model	#Parsable	Avg. #Diag.
Qwen2 7B	0.00%	-
Llama 3.1 8B	0.00%	-
Mistral 7B	0.00%	-
Mistral Nemo	0.00%	-
Qwen2 72B	0.00%	-
Llama 3.1 70B	0.00%	-
Qwen2 7B Instruct	97.25%	19.91
Llama 3.1 8B Instruct	0.49%	-
Mistral 7B Instruct	99.76%	18.16
Mistral Nemo Instruct	0.07%	20.00
Qwen2 72B Instruct	100.00%	20.12
Llama 3.1 70B Instruct	3.82%	20.16

Table 8: Capability of evaluated models to produce valid JSON for the ICD-10 HOSP dataset in the zeroshot scenario (5493 examples). Non-instruct fine-tuned models all fail to generate valid JSON. Model size only has a small effect on the validity of the generated output in comparison to instruct-finetuning (Llama 3.1 vs Qwen2 or Mistral 7B vs Mistral Nemo).

Model	Recall	Precision	MAP
all-mpnet-base-v2	9.99	6.02	10.94
S-PubMedBERT MSMarco	10.61	6.31	10.71
NeuML PubMedBERT	**10.72**	**6.92**	**11.82**

Table 9: Average diagnosis prediction performance over all datasets and all experiments using different class mapping models in macro averaged metrics. NeuML PubMedBERT works best. However, performance differences between the different class mapping models are rather small.

B.2 Performance drop for longer inputs.

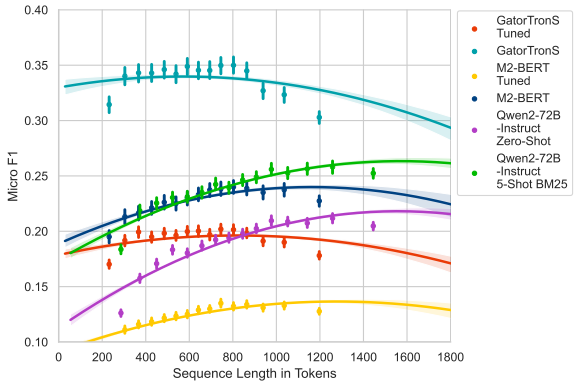

Figure 5: Micro F_1 over different sequence lengths averaged over all dataset splits. Encoder models have a reduced performance with sequences >850 tokens while the generative models tend have a more consistent performance on longer sequences. The token count varies across models due to differences in tokenizers.

In Fig. 5, we demonstrate that longer sequence inputs correlate with a decrease in micro F_1 performance. As anticipated, this decline is more pronounced for the encoder-based models BioMedBERT and GatorTronS, which are limited to sequence lengths of 512 tokens. Models that are capable of processing extended sequences like the chosen generative models and the M2 encoder, exhibit only slight decreases in performance on longer sequences. This highlights the necessity for long-context-capable domain-specific models.

B.3 ICU Results

Data	Retriever	#FS	Rec.	Prec.	MAP	MD Acc.
ICU	Zero-Shot prompted	0	**24.11**	14.20	16.82	48.11
	Random	1	17.67	13.95	16.06	40.80
		3	15.90	13.27	16.00	41.22
		5	14.24	13.45	15.77	40.66
	BM25	1	19.97	15.29	17.46	47.27
		3	18.89	**15.89**	19.09	51.18
		5	17.42	15.73	**20.48**	52.06
	PubMedBERT$_{NeuML}$	1	18.25	13.83	17.16	45.59
		3	17.13	13.97	18.43	50.50
		5	16.10	14.48	19.92	52.34
	BioMedBERT$_{ft}$	1	18.97	14.60	17.69	44.80
		3	18.04	14.86	18.96	50.92
		5	18.12	15.44	19.57	**52.59**
	Gold Heuristic	1	21.31	17.79	19.75	48.45
		3	22.72	20.24	26.88	59.15
		5	22.83	21.21	29.33	61.66

Table 10: Few-shot performance of Qwen2-72B-Instruct averaged over ICD-9 and ICD-10 ICU splits using the depicted retrieval models. Metrics are macro averaged. BioMedBERT$_{ft}$ is a custom fine-tuned encoder model on the respective training split of the dataset. (**#FS**: number of few-shot demonstrations, **MD Acc.**: Main diagnosis accuracy)

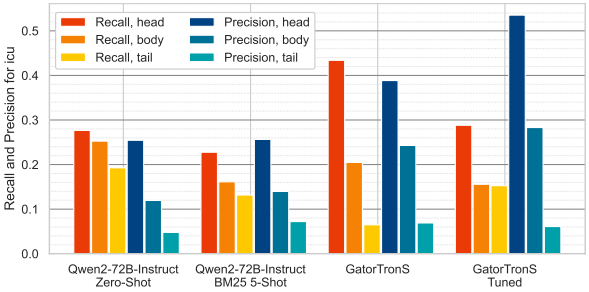

Figure 6: Macro recall and precision by model and classes grouped into tertiles by class frequency aggregated over both ICU splits (ICD-9 and ICD-10).

Data	Model	Zero-Shot prompted					Zero-Shot prompted + CoT					Few-Shot 5 Demonstrations				
		Rec.	**Prec.**	**MAP**	**MD Acc.**	**F1**	**Rec.**	**Prec.**	**MAP**	**MD Acc.**	**F1**	**Rec.**	**Prec.**	**MAP**	**MD Acc.**	**F1**
ICU	**Encoder-based Models**															
	BioMedBERT-110M	19.60	17.99	32.44	72.83	22.00			–					–		
	BioMedBERT-110M-tuned	15.56	23.13	20.03	55.91	20.71			–					–		
	GatorTronS-345M	**28.61**	23.96	**35.66**	**78.44**	**25.77**			–					–		
	GatorTronS-345M-tuned	20.00	**29.62**	21.13	63.66	25.34			–					–		
	M2-80M	10.16	08.27	23.12	56.05	9.50			–					–		
	M2-80M-tuned	08.13	08.98	10.72	35.05	8.34			–					–		
	Generative LLM															
	Llama 3.1-8B	13.97	14.81	13.50	33.43	11.20	12.25	13.41	12.36	35.57	09.76	14.12	14.60	17.62	48.33	13.20
	Llama 3.1-8B-Instruct	19.59	13.93	15.63	47.34	13.02	17.88	13.00	13.21	47.65	11.49	15.52	14.36	15.93	44.27	13.32
	Llama 3.1-70B	15.40	12.44	12.30	33.60	10.48	18.73	11.87	12.98	40.81	11.22	13.74	12.10	16.54	50.85	12.13
	Llama 3.1-70B-Instruct	20.31	13.36	14.70	50.37	12.55	21.24	12.66	15.85	50.18	12.08	17.06	15.38	16.24	44.90	14.47
	Mistral-7B	08.84	13.49	09.94	25.66	08.08	00.34	02.63	04.58	01.38	00.12	11.08	11.92	14.40	43.16	10.79
	Mistral-7B-Instruct	23.65	13.67	16.86	51.37	13.69	21.59	12.12	14.64	46.10	12.30	20.96	15.08	16.98	47.72	14.96
	Mistral-Nemo-13B	13.49	14.31	13.55	31.63	10.86	11.11	11.63	10.86	37.51	08.23	11.31	12.73	15.13	44.95	11.00
	Mistral-Nemo-13B-Instruct	19.64	11.97	13.87	48.39	11.58	19.03	11.96	12.46	48.54	10.78	11.24	13.35	15.24	43.28	11.37
	Qwen2-7B	20.95	13.92	15.54	40.97	13.11	19.05	13.96	15.49	40.19	12.55	11.23	13.36	15.80	43.35	11.35
	Qwen2-7B-Instruct	23.06	14.14	15.33	45.86	14.10	22.13	13.89	15.18	46.30	13.64	18.23	14.40	17.61	50.52	14.15
	Qwen2-72B	22.55	14.06	15.89	43.28	13.70	20.32	13.14	15.57	40.93	12.56	15.27	13.80	17.39	49.40	13.31
	Qwen2-72B-Instruct	24.11	14.20	16.82	48.11	14.41	22.78	14.10	16.47	47.16	14.12	17.42	15.74	18.75	52.06	14.77

Table 11: (**MAP**: *Mean Average Precision*, **MD Acc.**: *Main Diagnosis Accuracy*) Zero-shot performance on the ICU split over the top-20 predictions. The reported metrics are macro averaged. We aggregate ICD-9 and ICD-10 and report an unweighted mean. The dashed lines separate the encoder from the generative LLMs. CoT is not beneficial while instruct fine-tuning boosts performance. Threshold tuning decreases the performance of encoder models within the top-20 predictions. We depict the best performing models in bold and underline the best performing generative models in each category.

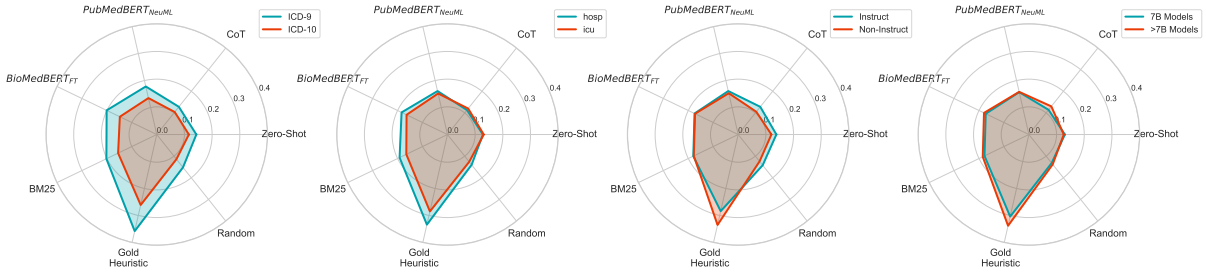

Figure 7: MAP for all generative models (Zero-Shot, CoT, and 5-shot with the respective retriever) aggregated for following criteria: Type of ICD codes, tasks, instruction and model size.

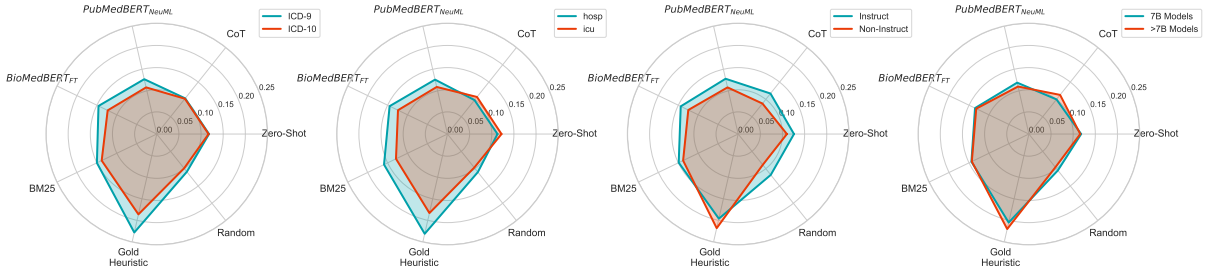

Figure 8: Macro F1 for all generative models (Zero-Shot, CoT, and 5-shot with the respective retriever) aggregated for following criteria: Type of ICD codes, tasks, instruction and model size.

Figure 9: Macro recall for all generative models (Zero-Shot, CoT, and 5-shot with the respective retriever) aggregated for following criteria: Type of ICD codes, tasks, instruction and model size.

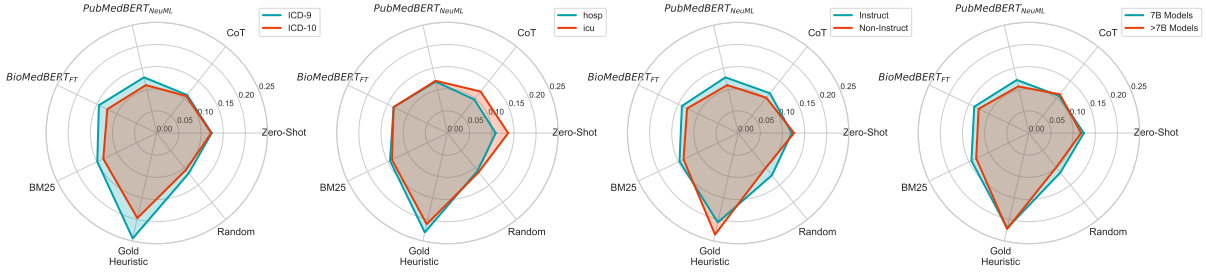

Figure 10: Macro precision for all generative models (Zero-Shot, CoT, and 5-shot with the respective retriever) aggregated for following criteria: Type of ICD codes, tasks, instruction and model size.