LLMs can construct powerful representations and streamline sample-efficient supervised learning

Ilker Demirel MIT
Lawrence Shi MIT
Zeshan Hussain MIT, Harvard Medical School
David Sontag MIT

Abstract

As real-world datasets become more complex and heterogeneous, supervised learning is often bottlenecked by input representation design. Modeling multimodal data, such as time-series, free text, and structured records, often requires non-trivial domain expertise. We propose an agentic pipeline to streamline this process. First, an LLM analyzes a small but diverse subset of *text-serialized* input examples *in-context* to synthesize a *global rubric*, which acts as a programmatic specification for extracting and organizing evidence. This rubric is then used to transform *naive* text-serializations of inputs into a more standardized format for downstream models. We also describe *local rubrics*, which are task-conditioned summaries generated by an LLM. Across 15 clinical tasks from the EHRSHOT benchmark, our rubric-based approaches significantly outperform traditional count-feature models, naive LLM baselines, and a clinical foundation model pretrained on orders of magnitude more data. Beyond performance, rubrics offer operational advantages such as being easy to audit, cost-effectiveness at scale, and facilitating tabular representations. For code and project website, visit https://lrrlpaper.github.io.

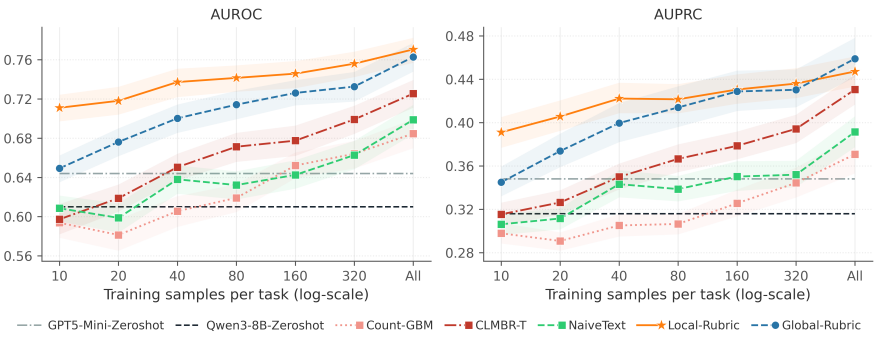


Figure 1: Performance averaged over all 15 clinical prediction tasks in the EHRSHOT benchmark [68]. Our rubric-style representations constructed by LLMs outperform the *naive* text-serialization baseline in Hegselmann et al. [15], as well as a clinical foundation model pretrained on 2.57M patients (CLMBR-T, [68]), and a count feature-based gradient boosting machine (Count-GBM, [23, 68]).

1 Introduction

Supervised learning underpins a wide range of applications across domains. In medicine, deep neural networks achieve specialist-level performance in pneumonia detection and diabetic retinopathy screening [12, 44]. In finance, credit risk assessment models outperform legacy scorecards [28]. In environmental science, supervised learning enables weather forecasting from radar observations [45].

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

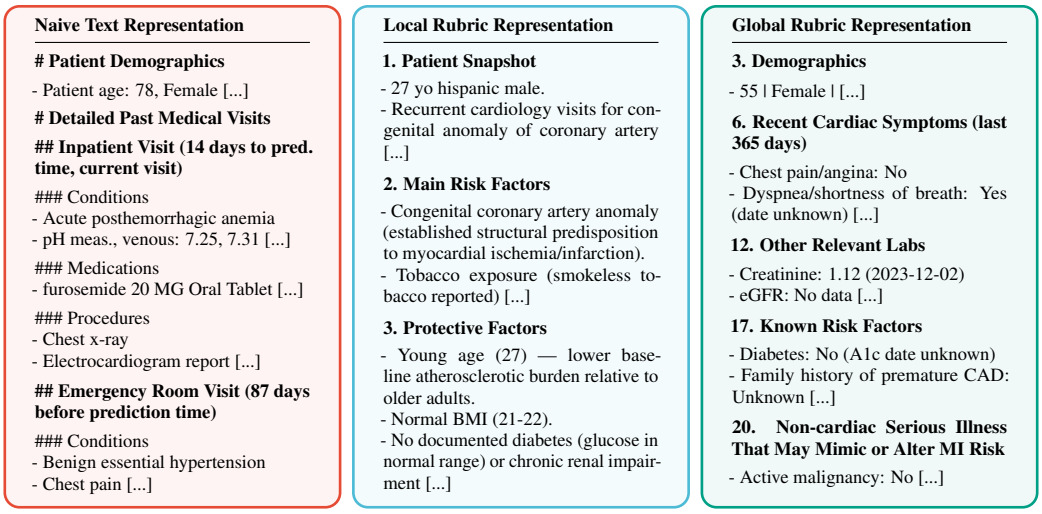


Figure 2: Synthetic electronic health record representation examples. *Left.* Naive text-serialization adopted from Hegselmann et al. [15]. *Middle.* Local rubric representation (task-conditioned summary of the naive text-serialization). *Right.* Global rubric transformed version of the naive-text serialization.

1.1 Representational Challenges in Complex Domains

The success of supervised learning depends on the availability of input representations that can be easily processed by off-the-shelf models. Real-world datasets, however, are increasingly more complex and heterogeneous. They combine structured fields with unstructured text, time-series, and images. In healthcare, clinical prediction may benefit from longitudinal labs, coded events (*e.g.*, diagnoses), free-text notes, and medical images. In finance, stock-price forecasting and risk modeling may involve price and volume time-series, news reports, and structured events such as analyst ratings.

In domains with complex data, representation design requires bespoke engineering and domain expertise. Even then, resulting representations are not necessarily optimal: they may discard critical signal or bury it in noise. We show how large language models (LLM) can build *agentic* pipelines that automate designing powerful input representations and enable sample-efficient supervised learning.

1.2 LLMs as a Representation Layer

LLMs offer a practical interface to heterogeneous data through text-serializations. Song et al. [54] and Akhauri et al. [3] serialize diverse configurations and logs into text sequences to predict system performance metrics. Hegselmann et al. [15] serialize longitudinal electronic health records (EHR) into Markdown and train linear heads over text-embeddings for clinical prediction tasks (see Figure 2, *left*). Demirel et al. [8] use LLMs to predict daily user activities by combining multimodal time-series data from wearables after transforming them into textual descriptions.

These works show LLMs' potential to streamline supervised learning with complex datasets, but they treat text-serialization of the input as fixed and leave the bulk of *calibration* to data for the downstream model. In contrast, we take the text-serialized input as a starting point and show how LLMs can automate constructing better representations to directly improve downstream performance.

1.3 Pretraining Knowledge for Sample-efficiency

Beyond modeling convenience, LLMs' pretraining knowledge can enable effective regularization, which is key to sample-efficiency. Our work aligns with literature on injecting knowledge into statistical models: LMPriors uses language descriptions as task-specific priors [7], and TabLLM shows effective few-shot tabular learning [14]. LLM-Select and LLM-Lasso guide feature selection and regularization [18, 74], and Kim et al. [25] use task metadata to construct inductive biases.

These methods mostly use LLMs to augment traditional models working on clean datasets. In contrast, we focus on representation design: how complex inputs should be organized *prior* to downstream

learning. In that sense, our work also aligns with recent literature on learning in the language space, such as GEPA by Agrawal et al. [1]. An extended related work section is included in Appendix A.

1.4 Our Contribution: Rubric Representation Learning

We propose *rubrics*, which are used to process complex text inputs into a standardized and information-rich format that can be easily and efficiently digested by downstream learners. We assume *naive* text-serializations of inputs are available, or they can be constructed straightforwardly (see Figure 2, left for an example). We develop two types of rubrics, which are given below and detailed in Section 2.

Global Rubrics. A global rubric is a task-level specification that defines what information should be extracted from the input and how. It is *generated* by prompting an LLM with a diverse set of examples (see Figure 3, Panel C).

Local Rubrics. We prompt an LLM to produce a *task-conditioned* local summary with structured sections (see Figure 4, left), similar to recent work on explainable clinical prediction models [42].

Advantages of Global Rubrics. Both rubrics achieve similar downstream performance and outperform the baselines. However, local rubrics do not have the same level of standardization as global rubrics, which endows the latter with several practical desiderata lacking in the former.

- *Auditable and improvable:* Global rubrics are more amenable to inspections by domain experts, such as for analyzing subgroup bias risk and iterative refinement.
- *More operationally useful:* Global rubric representations can be transformed into tabular features (Figure 3, Panel F), immediately unlocking a suite of machine learning techniques.
- *Cheaper to deploy:* Global rubric transformation at inference time can be automated (see Figure 3, Panels E and F), whereas summarization requires an LLM forward pass per example. This makes global rubrics "free" compared to local rubrics, which incur $\mathcal{O}(N)$ time and compute cost.

1.5 Evaluation in the EHRSHOT Benchmark: Clinical Outcome Prediction

We evaluate on 15 binary clinical prediction tasks from the EHRSHOT benchmark [68], spanning operational outcomes, new diagnoses, lab results, and chest X-ray findings. We compare against a gradient boosting machine with count-based features (Count-GBM, [23]), a clinical foundation model pretrained on 2.57M patients (CLMBR-T, [68]), zero-shot chain-of-thought prompting (CoT) with Qwen3-8B and GPT5-mini[1] [39, 43, 66], and the LLM baseline of Hegselmann et al. [15] using naive EHR text serializations. Our rubric-based representations substantially outperform baselines on average across different training sample sizes, particularly in the small-sample regime.

2 Rubric Representation Learning with LLMs

2.1 Global Rubrics

We introduce global rubrics for converting heterogeneous, weakly structured inputs into standardized, task-aligned representations. While our experiments focus on electronic health records (EHR), the procedure can be applied wherever inputs can be converted to a textual representation.

Setup and Notation. We describe the global rubric learning procedure for a single prediction task. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ denote labeled training data, where x is a *raw input* and $y \in \{0, 1\}$ is the task label. Let $s(\cdot)$ be some serialization procedure that maps an input x into its textual representation, and define $x^{\text{text}} = s(x)$. A rubric specifies a task-specific transformation

$$\mathcal{R} : \; x^{\text{text}} \mapsto x^{\text{rubric}},$$

where x^{rubric} is a more structured representation of the same underlying input x, and it can be used with downstream predictors instead of x^{text}. We describe downstream training in Section 4.

[1] We used GPT5-mini and GPT-5.2 via the HIPAA-compliant Microsoft Azure OpenAI Service.

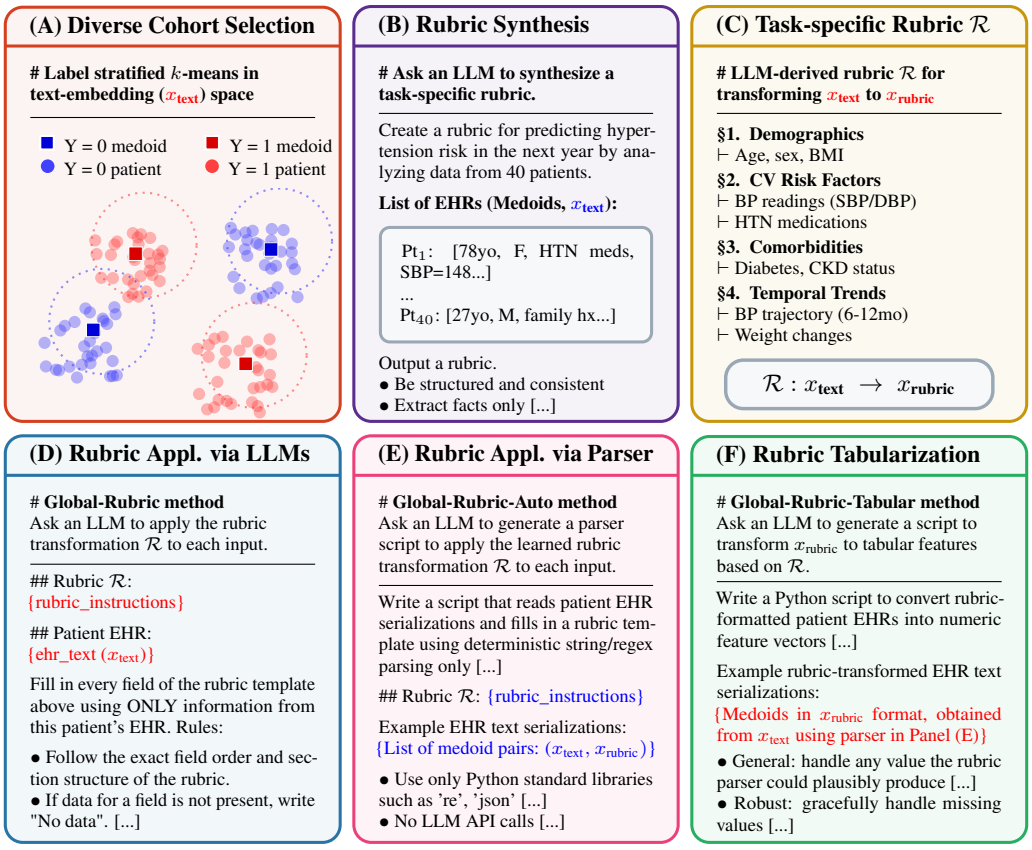


Figure 3: Agentic global-rubric pipeline for EHRSHOT tasks. Full prompts are in Appendix D.

Global Rubric Synthesis. Global rubric learning has two stages, shown in Panels A and B of Figure 3. First, we select a small, label-balanced and *diverse* cohort from the training split. Second, an LLM inspects this cohort *in-context* and synthesizes a task-specific rubric by defining predictive features and describing how to extract them.

- *Step 1) Diverse cohort selection:* Rubric synthesis is done through a single prompt to an LLM (GPT5-Mini). We first embed each text-serialized input x_i^{text} into a vector space using a pretrained text-embedding model and stratify by label,

$$\mathcal{D}^+ = \{x_i^{\text{text}} : y_i = 1\}, \qquad \mathcal{D}^- = \{x_i^{\text{text}} : y_i = 0\}.$$

  We then perform k-means clustering independently within each stratum ($\mathcal{D}^+$ and $\mathcal{D}^-$) in the embedding space. Here, k denotes the number of clusters *per label stratum*, so the cohort contains $2 * k$ examples in total. Due to context windows limitations, we use $k = 20$. Finally, we select the medoid from each cluster to obtain a diverse cohort.

- *Step 2) Rubric synthesis:* Given the selected cohort, we ask an LLM (GPT5-Mini) to produce a task-specific rubric that (i) defines discriminative, task-relevant signals, and (ii) specifies how each signal should be extracted from a given input, x^{text}. The full rubric-synthesis prompt is provided in Appendix D.1, and two full-rubric examples can be found in Appendix G.

Global Rubric Application. Once a global rubric $\mathcal{R}$ is learned, we apply it to naive text-serialization of each input, x^{text}, to produce x^{rubric}. We propose three different methods for the rubric transformation step, each with their own advantages. Full prompts are in Appendix D.

- *Rubric application via an LLM (Figure 3, Panel D):* We prompt an LLM (GPT5-Mini) with the global rubric $\mathcal{R}$, and the naive text-serialization of the input x^{text}, asking it to return x^{rubric}.

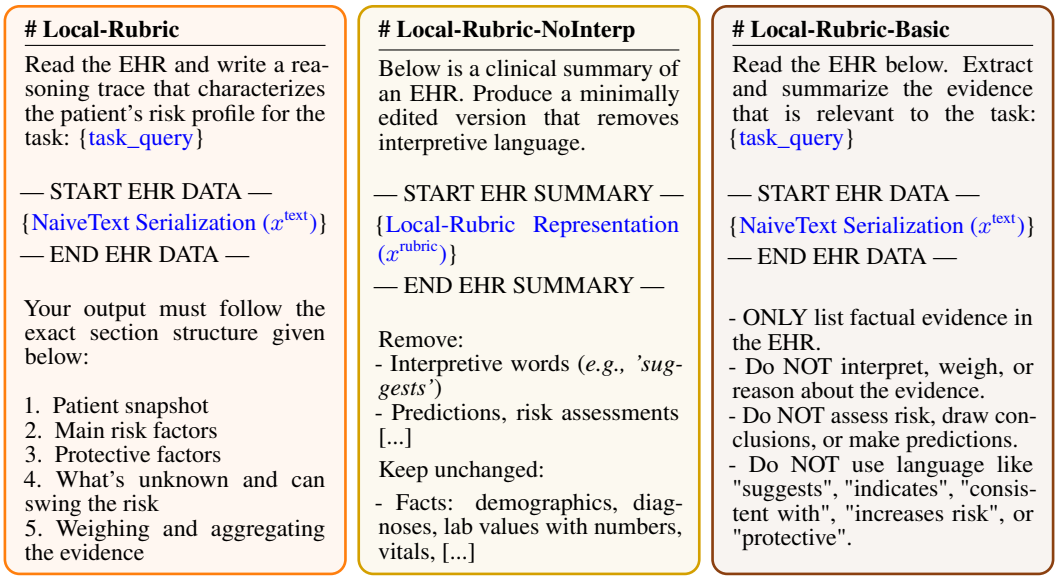


Figure 4: Prompts for generating local rubric representations. *Left.* Task-conditioned interpretive summaries (Local-Rubric). *Middle.* Edited summaries with interpretations stripped away (Local-Rubric-NoInterp, full prompt in Appendix D.5). *Right.* Fact-listing summaries (Local-Rubric-Basic).

- *Rubric application via a parser (Figure 3, Panel E):* We prompt an LLM (GPT-5.2) with the global rubric $\mathcal{R}$, and some paired examples (40) of x^{text}, and corresponding rubric representations, x^{rubric}. We ask the LLM to write a *deterministic* parser script to convert x^{text} to x^{rubric}.

- *Rubric-based tabularization (Figure 3, Panel F):* We prompt an LLM (GPT-5.2) with the global rubric $\mathcal{R}$, parser script for the rubric transformation (see item above), and some examples (40) of *parser-generated* rubric-transformations, x^{rubric}. We ask the LLM to write a script to convert x^{rubric} into a set of tabular features.

2.2 Local Rubrics

Global rubrics define a structure shared across *all* inputs after the rubric transformation. Beyond performance gains, this shared structure unlocks several practical and methodological advantages, as summarized in Section 1.4. However, it is important to characterize how much of the gain comes from standardizing the input representation versus the LLM's ability to augment the representation in the *language space* using pretrained world knowledge.

To study this, we introduce *local rubrics*: task-conditioned summaries of x^{text}, generated independently for each example by an LLM (GPT5-Mini). Local rubrics do not impose as rigid a template as global rubrics, aside from a general section structure. This gives the LLM flexibility to extract and interpret task-relevant evidence for each individual case in a more tailored way, ensuring no critical information is discarded. We discuss how this supports sample-efficiency in Section 5 (see Figure 1). The prompt used to generate the *Local-Rubric* representations is shown in Figure 4, left.

To further isolate the role of pretrained knowledge in sample-efficiency, we include two ablations that progressively remove it from the *Local-Rubric* representation. First, *Local-Rubric-NoInterp* starts from *Local-Rubric* and removes explicit interpretive or predictive language while preserving factual evidence (see Figure 4, middle). Because it is derived from *Local-Rubric*, it can still retain cues such as pointers to missing or potentially important unobserved information. Second, *Local-Rubric-Basic* asks the LLM to summarize only task-relevant facts, without weighing evidence, reasoning, or assessing risk (see Figure 4, right). It is therefore the simplest of the three summaries, with the least amount of LLM world knowledge encoded in x^{rubric}, and may omit useful cues retained by *Local-Rubric-NoInterp*, such as missing-information signals.

3 EHRSHOT Benchmark

We evaluate on EHRSHOT [68], a longitudinal EHR benchmark which contains deidentified data from 6,739 patients at Stanford Medicine, including demographics, diagnoses, procedures, medications, and labs across full patient timelines with millions of coded events. Wornow et al. [68] also release CLMBR-T, an autoregressive transformer pretrained on 2.57M patients for next-code prediction.

Clinical Prediction Tasks. EHRSHOT comprises 15 binary classification tasks across 4 categories:

- *Operational outcomes*: ICU transfer, Long length of stay ($>$ 7 days), 30-day readmission
- *Assignment of new diagnoses*: Acute myocardial infarction (MI), Celiac, Hyperlipidemia, Hypertension, Lupus, Pancreatic cancer
- *Anticipating labs*: Anemia, Hyperkalemia, Hypoglycemia, Hyponatremia, Thrombocytopenia
- *Chest X-ray*: Abnormal chest x-ray findings

Operational tasks predict near-term events in the context of the current visit. Diagnosis tasks predict new diagnoses within one year. Lab tasks predict abnormal upcoming results for the most recent lab order. The chest X-ray task predicts abnormal radiology findings.

Subsampling. As some methods require LLM calls per example, we subsample the EHRSHOT dataset for budget and computational reasons. Details are provided in Appendix B.

Evaluation. Models are trained on the training split, with their hyperparameters tuned based on the validation split, and evaluated on the test split. We report AUROC and AUPRC, which are more appropriate than threshold-dependent metrics such as F1-score [61], along with 95% confidence intervals obtained by bootstrapping in the test split.

4 Methods and Downstream Training

4.1 Our Rubric Representation-based Methods

In Section 2, we proposed the global and local rubrics. Here, we describe downstream training for those. Six of the rubric representations produce textual patient-level representations for x^{rubric}:

- Global-Rubric: The rubric $\mathcal{R}$ is generated by an LLM after inspecting a small set of input examples in context. Another LLM then applies $\mathcal{R}$ to convert x^{text} into x^{rubric} (Figure 3, Panel D).
- Global-Rubric-Blind: The rubric $\mathcal{R}$ is generated without input examples, using only the task description and LLM's world knowledge. Another LLM applies $\mathcal{R}$ to convert x^{text} into x^{rubric}.
- Global-Rubric-Auto: Same rubrics as in Global-Rubric are used, but an LLM first generates a parser script that deterministically converts x^{text} into x^{rubric} according to $\mathcal{R}$ (Figure 3, Panel E).
- Local-Rubric: An LLM directly converts each x^{text} into a task-conditioned patient summary x^{rubric}, including both relevant evidence and its task-specific interpretation (Figure 4, left).
- Local-Rubric-NoInterp: Starting from the Local-Rubric representation, we remove the explicit interpretation of risk from x^{rubric}, and retain only the extracted factual evidence (Figure 4, middle).
- Local-Rubric-Basic: An LLM converts each x^{text} into x^{rubric} using a prompt that asks only for task-relevant facts and evidence, without any reasoning or interpretation (Figure 4, right).

For all methods above, the resulting textual x^{rubric} is converted into a numerical vector by passing it through a pretrained text-embedding model. We report results for `Qwen3-8B-Embedding` [77] in the main paper, with additional embedding models, `LLaMA-3-8B` [5, 36], `Mistral-7B` [5, 19], and OpenAI's `text-embedding-3-large` [38], reported in Appendix C.5. We then fit logistic regression classifiers on top of these vector embeddings. Finally, we also evaluate a tabular variant:

- Global-Rubric-Tabular: An LLM generates a parser script that converts the global rubric output into tabular features (Figure 3, Panel F), which are then used to fit an XGBoost classifier [6].

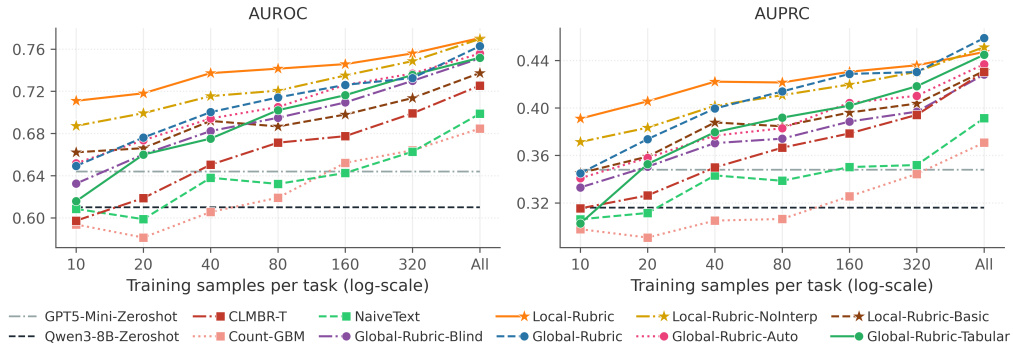


Figure 5: Average performance over all tasks. For confidence bands, see Appendix C.2.

4.2 EHRSHOT Baselines

We include Count-GBM following EHRSHOT [68], where each EHR is converted into a vector of code counts observed prior to the prediction time. Different time-windows are used, such as last 90 days and 90-180 days before. A LightGBM classifier is then trained [23].

We also evaluate CLMBR-T, a transformer-based autoregressive medical foundation model. It is pretrained with a next-code prediction objective, using longitudinal data from 2.57M patients drawn from the same distribution as the EHRSHOT dataset [68]. For downstream tasks, a logistic regression classifier is fit on top of the vector embeddings extracted from CLMBR-T.

4.3 LLM-based Baselines

Our rubric representations, x^{rubric}, are derived from *naive* text-serializations of the input, x^{text}. We adopt the serialization introduced by Hegselmann et al. [15] (see Figure 2, left). Each record includes patient demographics, a "General Medical Events" section for codes that are not tied to any visit, and a "Detailed Past Medical Visits" section listing visits in reverse chronological order. We refer to the baseline that uses x^{text} directly as NaiveText. Same as textual rubrics, x^{text} is embedded using a pretrained embedding model, and a logistic regression classifier is trained on top.

We also evaluate zero-shot chain-of-thought (CoT) prompting [66]. For each example, Qwen3-8B and GPT5-Mini are prompted to reason over the NaiveText EHR serialization and give a final `Yes/No` answer. We sample 10 responses and estimate the probability as the fraction of `Yes` answers.

5 Quantitative Results

Rubrics Dominate the Baselines on Average. We evaluate the 15 EHRSHOT tasks across training-set sizes from n=10 per class to n=All (Figures 5 and 6, with numerical results in Table 2 in Appendix C.1, and per-task tables in Appendix C.6). We use 1:1 label-balanced training sets except for n=All. Global-Rubric and Local-Rubric outperform every baseline for all values of n, with the largest gap at small n. At n=10, Global-Rubric reaches 0.649 AUROC (0.345 AUPRC) and Local-Rubric 0.701 (0.382), while NaiveText, CLMBR-T, and Count-GBM all remain near 0.60 (0.30). At n=All, Global-Rubric obtains 0.763 (0.459) and Local-Rubric 0.772 (0.452), with the strongest baseline, CLMBR-T, trailing at 0.725 (0.430). On rare-event tasks, rubric methods roughly double the baselines' AUPRC (e.g., new diagnoses at n=10).

Sample-efficiency: Gains are Largest at Small n. Rubric methods outpace traditional baselines most strongly when labels are scarce, and the gap narrows but does not close at n=All. CLMBR-T, in particular, trails our rubric methods by more than 0.10 AUROC and $\sim$ 0.07 AUPRC at n=10. CLMBR-T is pretrained on 2.57M patients, data that is not available to any other method, yet rubrics surpass its performance using only the training set.

Input Representation Design is Crucial to Performance Gains. Global-Rubric outperforms NaiveText by 0.04/0.06 AUROC and 0.04/0.07 AUPRC at n=10/All, and Local-Rubric by 0.09/0.07

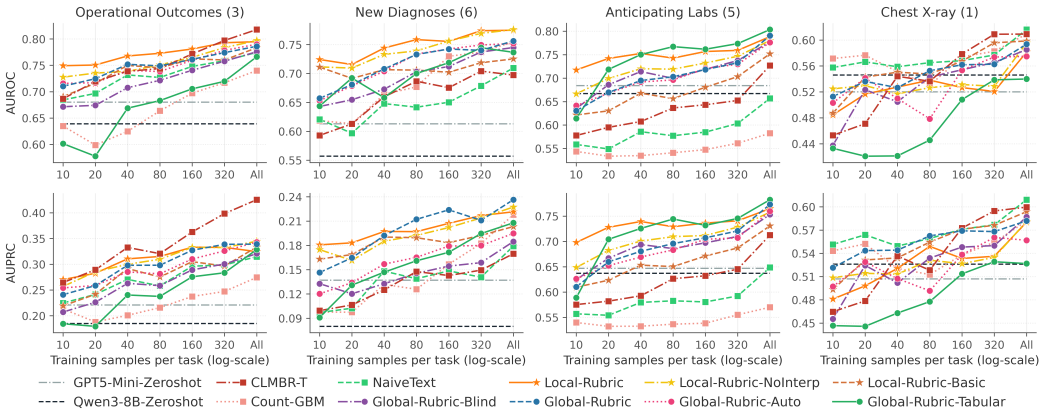


Figure 6: Average performance by task group. For confidence bands, see Appendix C.2.

AUROC and 0.08/0.06 AUPRC. NaiveText and our rubric methods share the same downstream stack, an LLM-based pretrained text-embedding model with a logistic-regression head. The gap to NaiveText therefore isolates the value of input representation design prior to downstream learning, performed by LLMs in the language space via rubrics.

Local vs. Global Rubrics. Local-Rubric leads at n=10 (AUROC 0.701 vs 0.649, AUPRC 0.382 vs 0.345), but the two converge at n=All, with Global-Rubric slightly ahead in overall AUPRC (0.459 vs 0.452). Local rubrics are tailored per patient, so the LLM has the flexibility to emphasize the most informative evidence. In contrast, global rubrics impose a fixed template on every patient and may therefore be noisier when fields are not as relevant for some patients, or fail to represent some important information about a particular patient. Further, Local-Rubric representations contain reasoning and assessment based on the evidence through LLMs' world knowledge, which is lacking from global rubrics, hinting at the value of LLMs' pretrained knowledge for sample-efficiency. Convergence of Global-Rubric and Local-Rubric performance for large n suggests that the former is a noisier representation rather than systematically missing information.

Ablations Further Support the Value of Pretraining Knowledge for Sample-efficiency. Stripping interpretive language from the Local-Rubric representation while retaining factual evidence and pointers to potentially missing or conflicting evidence (Local-Rubric-NoInterp) costs some performance at n=10 but is essentially neutral at n=All, showing how injecting LLMs' pretrained knowledge into input representations helps with sample-efficiency. An even simpler summary that only lists facts (Local-Rubric-Basic) costs more performance, and the gap remains visible at n=All.

Global Rubric Variants: The Cost of Automation. Global-Rubric-Auto, which replaces per-example LLM calls with a deterministic parser, trails Global-Rubric closely across sample sizes, suggesting that rubric application can essentially be made free at deployment. In contrast, Global-Rubric-Tabular lags visibly at small n. We hypothesize that this is due to XGBoost's inability to leverage a prior, as opposed to the text-embedding models used in other methods. As expected, it nearly catches up at n=All, and on lab tasks it is in fact the single strongest method overall. Finally, Global-Rubric-Blind generates the rubric from the task description alone. It continues to outperform the baselines across sample sizes but remains inferior to Global-Rubric, showing the importance of giving the LLM a chance to look at some data first (see Panel A in Figure 3).

Operational Outcomes. CLMBR-T is the strongest here at n=All (0.818 AUROC, 0.425 AUPRC), edging out Local-Rubric (0.802, 0.341) and Global-Rubric (0.786, 0.339), with a more pronounced lead in AUPRC. We attribute this to alignment between CLMBR-T's next-code pretraining and operational targets. For instance, every visit in the pretraining corpus contributes an implicit "label" for the ICU admission task via next-code prediction, which is not the case for other tasks such as lab results. Even so, in the small-sample regime our rubrics overtake CLMBR-T in AUROC (Local-Rubric 0.750, Global-Rubric 0.710 at n=10, vs CLMBR-T 0.687) and match it in AUPRC.

A. Preparation (before extracting)
- Define time windows: *Very recent*: last 30 days, *Recent*: 31-180 days, *Baseline/remote*: >180 days
- Standardize units: *Blood pressure*: mmHg (systolic/diastolic), *Weight*: kg or oz → convert to kg [...]

Step 2 - Blood pressure (BP) data extraction and normalization
- Extract systolic/diastolic BP values with timestamps and context (office, inpatient, ED, home, ambulatory, perioperative).
- For each time window (very recent, recent, baseline): compute count, mean, median, SD, min, max; flag highest recent BP
- Compute simple trend metrics (e.g., recent slope; BP variability via SD).
- Categorize BP per ACC/AHA categories using aggregated recent values: Normal (<120/<80), Elevated (120-129/<80) [...]

Step 9 - Synthesis per domain (structured fields and scoring)
- For each domain, record presence, supporting data, recency, and confidence (High/Moderate/Low).
- *Domain A* - BP phenotype: last BP (date/context), mean recent BP (last 30d; 31-180d), BP category, variability flag, ambulatory BP.
- *Domain B* - Metabolic / vascular risk: Diabetes (Y/N) - last A1c (% and date), BMI and obesity category, Hyperlipidemia (Y/N) - LDL value and date, Smoking (current/former/never). Create a simple domain scorecard: number of High/Moderate/Minor risk features [...]

Figure 7: Key excerpts from learned hypertension global rubric instructions.

Lab Results. Rubric methods deliver the largest gains here. At n=All, Global-Rubric reaches 0.791 AUROC (0.773 AUPRC), Global-Rubric-Tabular 0.804 (0.783), and even Global-Rubric-Blind (0.777, 0.753) beats CLMBR-T (0.727, 0.713). The lab rubrics (Appendix G) make recent labs, electrolyte panels, volume status, renal function, and lab-affecting medications explicit, the kind of structured, recency-aware signal that is ordinarily diluted in raw EHR serializations.

New Diagnoses and Chest X-ray. Rubrics dominate the new-diagnosis group at small n (Local-Rubric 0.713/0.170 AUROC/AUPRC vs Count-GBM 0.619/0.098 at n=10), roughly 1.5–2× above baseline AUPRC and especially strong on rare-event tasks (Tables 13 and 14). Chest X-ray is the noisiest task in EHRSHOT. The binary label collapses an originally 14-way radiology label, and many patients have their last documented visit more than a month before the prediction time. All methods cluster near 0.60 AUROC at n=All, suggesting a ceiling on learnable signal rather than a method-specific failure.

Robustness Across Text-embedding Backbones. The rubric-over-NaiveText ordering is preserved with Mistral-7B, LLaMA-3-8B (LLM2Vec), and OpenAI's text-embedding-3-large (Appendix C.5), in both AUROC and AUPRC.

Additional Analyses and Generalization. Appendix H explores alternative downstream learning strategies, including LoRA fine-tuning and chain-of-thought (CoT) reasoning. In an earlier four-task evaluation, LoRA fine-tuning achieves competitive performance, while supervised CoT generally underperforms, motivating our simpler and more computationally efficient embedding-based pipeline. Appendix I evaluates Global-Rubric-Tabular beyond healthcare on online recruitment fraud detection, where it outperforms a hand-engineered expert feature set across training-set sizes, reaching 0.974 AUROC and 0.841 AUPRC with the full training set (vs. 0.934 and 0.616). Finally, Appendix J investigates potential data leakage using a likelihood-based membership probe, finding near-chance discrimination between real and perturbed EHRSHOT records (AUROC 0.57 and 0.52), providing reassuring, though not conclusive, evidence against memorization.

6 Qualitative Global Rubric Analyses: Future Hypertension Prediction

We examine the learned global rubric for the new hypertension diagnosis prediction task. An excerpt is shown in Figure 7, with the full rubric in Appendix G.1.

Temporal Standardization and BP Normalization. The rubric first imposes structure on noisy EHR data before extracting features. In the preparation stage (Figure 7, A), it defines explicit time windows: very recent ($\leq$ 30 days), recent (31–180 days), and baseline/remote ($>$ 180 days), and

standardizes units and formats, including blood pressure (BP) in mmHg and weight in kg. This is important for hypertension prediction because recency, persistence, and measurement comparability are central to distinguishing sustained hypertension risk from isolated or context-specific elevations.

Clinically Grounded BP Feature Construction. Step 2 converts raw BP readings into structured longitudinal features. The rubric first extracts systolic/diastolic values with timestamps and clinical context. Within each temporal window, it computes summary statistics such as mean and maximum. It also derives simple trend and variability measures such as recent slope and standard deviation (SD). Finally, it maps aggregated recent BP values to ACC/AHA categories, converting irregular raw measurements into clinically interpretable BP phenotype features [67].

Domain-level Synthesis. Step 9 further compresses the extracted evidence into domain-level summaries and a scorecard. For each domain, the rubric records presence, supporting data, recency, and confidence. Domain A summarizes BP phenotype, including last BP, recent means, BP category, variability, and ambulatory BP; Domain B captures metabolic and vascular risk factors such as diabetes, BMI/obesity, hyperlipidemia, LDL, and smoking. The resulting counts of high-, moderate-, and minor-risk features provide compact task-specific signals that can be used to effectively predict the cumulative hypertension risk across BP patterns and comorbidities.

More broadly, this highlights the value of global rubrics as a bridge between raw, heterogeneous inputs and downstream prediction. By making task-relevant evidence explicit and structured, rubrics provide models with more interpretable, standardized, and prediction-aligned representations. In Appendix F we examine the *learned tabular features* produced by Global-Rubric-Tabular for the hyponatremia lab task and show that they closely track the standard clinical decision tree for evaluating hyponatremia.

7 Concluding Remarks and Limitations

We studied how LLMs can construct powerful input representations for supervised learning in complex domains with heterogeneous inputs that are hard to model with off-the-shelf methods. Our central proposal, *rubric representation learning*, uses LLMs to transform naive text-serializations into task-aligned representations before downstream training. We introduced *global rubrics*, a shared structured template for extracting task-relevant evidence, and *local rubrics*, task-conditioned summaries for individual examples. Across 15 EHRSHOT prediction tasks, our rubric methods substantially outperformed the NaiveText baseline and surpassed CLMBR-T on average, where the latter is pretrained on 2.57M patients. Local-Rubric was strongest at small n, Global-Rubric matched or exceeded it at large n, and Global-Rubric-Tabular was the single strongest method on lab tasks at n=All. Input representation design is thus a first-order driver of downstream performance.

Beyond predictive performance, global rubrics impose a shared structure that makes them easier to audit and refine. Once learned from 40 examples, Global-Rubric-Auto and Global-Rubric-Tabular can be applied deterministically and on CPU, with deployment cost effectively $\mathcal{O}(1)$ versus $\mathcal{O}(N)$ LLM calls for local rubrics.

Limitations and Future Work. Our primary evaluation focuses on a single EHR benchmark, supplemented by an experiment on recruitment fraud detection (Appendix I), and does not include richer modalities such as clinical notes or imaging. Rubric learning is also less effective in some regimes, with CLMBR-T retaining an advantage on operational outcomes and chest X-ray remaining noisy across all methods. Rubric synthesis is currently bounded by context length, with each global rubric created from 40 patients; iterative refinement using additional examples, failure cases, or expert feedback is a natural next step. More broadly, LLMs may help automate one of the most persistent bottlenecks in applied machine learning: the design of effective input representations.

Acknowledgments and Disclosure of Funding

ID and DS were supported by Office of Naval Research Award No. N00014-21-1-2807. LS, ID, and DS were supported by a grant from Independence Blue Cross. The authors would also like to thank Shannon Shen from the ClinicalML group at MIT for insightful discussions.

References

[1] L. A. Agrawal, S. Tan, D. Soylu, N. Ziems, R. Khare, K. Opsahl-Ong, A. Singhvi, H. Shandilya, M. J. Ryan, M. Jiang, C. Potts, K. Sen, A. Dimakis, I. Stoica, D. Klein, M. Zaharia, and O. Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2026.

[2] M. Agrawal, S. Hegselmann, H. Lang, Y. Kim, and D. Sontag. Large language models are few-shot clinical information extractors. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 1998–2022, 2022.

[3] Y. Akhauri, B. Lewandowski, C.-H. Lin, A. N. Reyes, G. C. Forbes, A. Wongpanich, B. Yang, M. S. Abdelfattah, S. Perel, and X. Song. Performance prediction for large systems via text-to-text regression. *arXiv preprint arXiv:2506.21718*, 2025.

[4] S. Bedi, H. Cui, M. Fuentes, A. Unell, M. Wornow, J. M. Banda, N. Kotecha, T. Keyes, Y. Mai, M. Oez, et al. Holistic evaluation of large language models for medical tasks with medhelm. *Nature Medicine*, pages 1–9, 2026.

[5] P. BehnamGhader, V. Adlakha, M. Mosbach, D. Bahdanau, N. Chapados, and S. Reddy. LLM2Vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*, 2024. URL https://arxiv.org/abs/2404.05961.

[6] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

[7] K. Choi, C. Cundy, S. Srivastava, and S. Ermon. Lmpriors: Pre-trained language models as task-specific priors. *arXiv preprint arXiv:2210.12530*, 2022.

[8] I. Demirel, K. Thakkar, B. Elizalde, S. Y. Ren, and J. Narain. Using llms for late multimodal sensor fusion for activity recognition. In *NeurIPS 2025 Workshop on Learning from Time Series for Health*, 2025.

[9] S. L. Fleming, A. Lozano, W. J. Haberkorn, J. A. Jindal, E. Reis, R. Thapa, L. Blankemeier, J. Z. Genkins, E. Steinberg, A. Nayak, et al. Medalign: A clinician-generated dataset for instruction following with electronic medical records. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22021–22030, 2024.

[10] P. Garcia, S. P. Ma, S. Shah, M. Smith, Y. Jeong, A. Devon-Sand, M. Tai-Seale, K. Takazawa, D. Clutter, K. Vogt, et al. Artificial intelligence–generated draft replies to patient inbox messages. *JAMA Network Open*, 7(3):e243201, 2024.

[11] E. Goh, R. J. Gallo, E. Strong, Y. Weng, H. Kerman, J. A. Freed, J. A. Cool, Z. Kanjee, K. P. Lane, A. S. Parsons, et al. Gpt-4 assistance for improvement of physician performance on patient care tasks: a randomized controlled trial. *Nature Medicine*, 31(4):1233–1238, 2025.

[12] V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, R. Kim, R. Raman, P. C. Nelson, J. L. Mega, and D. R. Webster. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*, 316(22):2402–2410, 2016.

[13] S. Guo, C. Deng, Y. Wen, H. Chen, Y. Chang, and J. Wang. DS-agent: Automated data science by empowering large language models with case-based reasoning. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pages 16813–16848, 2024.

[14] S. Hegselmann, A. Buendia, H. Lang, M. Agrawal, X. Jiang, and D. Sontag. Tabllm: Few-shot classification of tabular data with large language models. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 5549–5581. PMLR, 2023.

[15] S. Hegselmann, G. von Arnim, T. Rheude, N. Kronenberg, D. Sontag, G. Hindricks, R. Eils, and B. Wild. Large language models are powerful electronic health record encoders. *arXiv preprint arXiv:2502.17403*, 2025.

[16] S. Hong, Y. Lin, B. Liu, B. Liu, B. Wu, C. Zhang, D. Li, J. Chen, J. Zhang, J. Wang, L. Zhang, L. Zhang, M. Yang, M. Zhuge, T. Guo, T. Zhou, W. Tao, R. Tang, X. Lu, X. Zheng, X. Liang, Y. Fei, Y. Cheng, Y. Ni, Z. Gou, Z. Xu, Y. Luo, and C. Wu. Data interpreter: An LLM agent for data science. In *Association for Computational Linguistics (ACL)*, pages 19796–19821, 2025.

[17] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations (ICLR)*, 2022.

[18] D. P. Jeong, Z. C. Lipton, and P. K. Ravikumar. LLM-select: Feature selection with large language models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

[19] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023. URL https://arxiv.org/abs/2310.06825.

[20] Y. Jiang, K. C. Black, G. Geng, D. Park, J. Zou, A. Y. Ng, and J. H. Chen. Medagentbench: a virtual ehr environment to benchmark medical llm agents. *Nejm Ai*, 2(9), 2025.

[21] D. Jin, E. Pan, N. Oufattole, W.-H. Weng, H. Fang, and P. Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.

[22] A. E. W. Johnson, T. J. Pollard, L. Shen, L.-W. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3:160035, 2016.

[23] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[24] J. Kim, A. Podlasek, K. Shidara, F. Liu, A. Alaa, and D. Bernardo. Limitations of large language models in clinical problem-solving arising from inflexible reasoning. *Scientific reports*, 15(1): 39426, 2025.

[25] J. Kim, C. Squires, and P. Ravikumar. Knowledge-enriched machine learning for tabular data. In *International Conference on Neuro-symbolic Systems*, pages 260–292, 2025.

[26] M. Kirchler, M. Ferro, V. Lorenzini, R. P. van de Water, F. G. A. 3, C. Lippert, and A. Ganna. Large language models improve transferability of electronic health record-based predictions across countries and coding systems. *npj Digital Medicine*, 2026.

[27] N. Lambert, J. Morrison, V. Pyatkin, S. Huang, H. Ivison, F. Brahman, L. J. V. Miranda, A. Liu, N. Dziri, S. Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

[28] S. Lessmann, B. Baesens, H.-V. Seow, and L. Thomas. Benchmarking state-of-the-art classification algorithms for credit scoring. *European Journal of Operational Research*, 247(1):124–136, 2015.

[29] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:9459–9474, 2020.

[30] Y. Liao, C. Wu, J. Liu, S. Jiang, P. Qiu, H. Wang, Y. Yue, S. Zhen, J. Wang, Q. Fan, et al. Ehr-r1: A reasoning-enhanced foundational language model for electronic health record analysis. *arXiv preprint arXiv:2510.25628*, 2025.

[31] J. Lin, Z. Wu, and J. Sun. Training llms for ehr-based reasoning tasks via reinforcement learning. *arXiv preprint arXiv:2505.24105*, 2025.

[32] C. Lu, C. Lu, R. T. Lange, J. Foerster, J. Clune, and D. Ha. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

[33] J. Mattern, F. Mireshghallah, Z. Jin, B. Schölkopf, M. Sachan, and T. Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison. In *Findings of the Association for Computational Linguistics: ACL 2023*, 2023.

[34] D. McDuff, M. Schaekermann, T. Tu, A. Palepu, A. Wang, J. Garrison, K. Singhal, Y. Sharma, S. Azizi, K. Kulkarni, et al. Towards accurate differential diagnosis with large language models. *Nature*, 642(8067):451–457, 2025.

[35] N. Mehandru, B. Y. Miao, E. R. Almaraz, M. Sushil, A. J. Butte, and A. Alaa. Evaluating large language models as agents in the clinic. *NPJ digital medicine*, 7(1):84, 2024.

[36] Meta AI. The Llama 3 herd of models. Technical report, Meta, 2024. URL https://arxiv.org/abs/2407.21783.

[37] J. Nam, J. Yoon, J. Chen, R. Sinha, J. Shin, and T. Pfister. DS-STAR: Data science agent for solving diverse tasks across heterogeneous formats and open-ended queries. *arXiv preprint arXiv:2509.21825*, 2025.

[38] OpenAI. New embedding models and API updates. https://openai.com/blog/new-embedding-models-and-api-updates, 2024. Accessed: 2025-06-15.

[39] OpenAI. Gpt-5 system card. https://cdn.openai.com/gpt-5-system-card.pdf, Aug. 2025.

[40] J. W. O'Sullivan, A. Palepu, K. Saab, W.-H. Weng, D. K. Amponsah, E. Cheng, Y. Cheng, E. Chu, Y. Desai, A. Elezaby, et al. A large language model for complex cardiology care. *Nature Medicine*, pages 1–8, 2026.

[41] A. Palepu, V. Dhillon, P. Niravath, W.-H. Weng, P. Prasad, K. Saab, R. Tanno, Y. Cheng, H. Mai, E. Burns, et al. Exploring large language models for specialist-level oncology care. *NEJM AI*, 2 (11):AIcs2500025, 2025.

[42] P. Petridis, G. Margaritis, V. Stoumpou, and D. Bertsimas. Holistic ai in medicine; improved performance and explainability. *npj Digital Medicine*, 2026.

[43] Qwen3Team. Qwen3 technical report, 2025.

[44] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M. P. Lungren, and A. Y. Ng. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017.

[45] S. Ravuri, K. Lenc, M. Willson, D. Kangin, R. Lam, P. Mirowski, S. Fitzpatrick, M. Athanassiadou, S. Kashem, S. Madge, R. Prudden, A. Mandhane, A. Clark, A. Brock, K. Simonyan, R. Hadsell, N. Robinson, E. Clancy, A. Arribas, S. Mohamed, and N. Kalchbrenner. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597:672–677, 2021.

[46] P. Renc, Y. Jia, A. E. Samir, J. Was, Q. Li, D. W. Bates, and A. Sitek. Zero shot health trajectory prediction using transformer. *NPJ digital medicine*, 7(1):256, 2024.

[47] S. Sandmann, S. Hegselmann, M. Fujarski, L. Bickmann, B. Wild, R. Eils, and J. Varghese. Benchmark evaluation of deepseek large language models in clinical decision-making. *Nature medicine*, 31(8):2546–2549, 2025.

[48] R. Shao, A. Asai, S. Z. Shen, H. Ivison, V. Kishore, J. Zhuo, X. Zhao, M. Park, S. G. Finlayson, D. Sontag, et al. Dr tulu: Reinforcement learning with evolving rubrics for deep research. *arXiv preprint arXiv:2511.19399*, 2025.

[49] W. Shi, A. Ajith, M. Xia, Y. Huang, D. Liu, T. Blevins, D. Chen, and L. Zettlemoyer. Detecting pretraining data from large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

[50] W. Shi, R. Xu, Y. Zhuang, Y. Yu, J. Zhang, H. Wu, Y. Zhu, J. C. Ho, C. Yang, and M. D. Wang. Ehragent: Code empowers large language models for few-shot complex tabular reasoning on electronic health records. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 22315–22339, 2024.

[51] C. Si, Z. Yang, Y. Choi, E. Candès, D. Yang, and T. Hashimoto. Towards execution-grounded automated ai research. *arXiv preprint arXiv:2601.14525*, 2026.

[52] K. Singhal, S. Azizi, T. Tu, S. S. Mahdavi, J. Wei, H. W. Chung, N. Scales, A. Tanwani, H. Cole-Lewis, S. Pfohl, et al. Large language models encode clinical knowledge. *Nature*, 620 (7972):172–180, 2023.

[53] K. Singhal, T. Tu, J. Gottweis, R. Sayres, E. Wulczyn, M. Amin, L. Hou, K. Clark, S. R. Pfohl, H. Cole-Lewis, et al. Toward expert-level medical question answering with large language models. *Nature medicine*, 31(3):943–950, 2025.

[54] X. Song, O. Li, C. Lee, B. Yang, D. Peng, S. Perel, and Y. Chen. Omnipred: Language models as universal regressors. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

[55] G. Spasovski, R. Vanholder, B. Allolio, D. Annane, S. Ball, D. Bichet, G. Decaux, W. Fenske, E. J. Hoorn, C. Ichai, et al. Clinical practice guideline on diagnosis and treatment of hyponatraemia. *Nephrology Dialysis Transplantation*, 29(suppl_2):i1–i39, 2014.

[56] E. Steinberg, J. A. Fries, Y. Xu, and N. Shah. MOTOR: A time-to-event foundation model for structured medical records. In *International Conference on Learning Representations (ICLR)*, 2024.

[57] T. Tu, M. Schaekermann, A. Palepu, K. Saab, J. Freyberg, R. Tanno, A. Wang, B. Li, M. Amin, Y. Cheng, et al. Towards conversational diagnostic artificial intelligence. *Nature*, 642(8067): 442–450, 2025.

[58] O. Unlu, J. Shin, C. J. Mailly, M. F. Oates, M. R. Tucci, M. Varugheese, K. Wagholikar, F. Wang, B. M. Scirica, A. J. Blood, et al. Retrieval-augmented generation–enabled gpt-4 for clinical trial screening. *NEJM AI*, 1(7), 2024.

[59] UnslothTeam. Unsloth, 2023. URL http://github.com/unslothai/unsloth.

[60] A. Upadhyay, B. L. Jaber, and N. E. Madias. Incidence and prevalence of hyponatremia. *The American journal of medicine*, 119(7):S30–S35, 2006.

[61] B. Van Calster, G. S. Collins, A. J. Vickers, L. Wynants, K. F. Kerr, L. Barreñada, G. Varoquaux, K. Singh, K. G. Moons, T. Hernandez-Boussard, et al. Evaluation of performance measures in predictive artificial intelligence models to support medical decisions: overview and guidance. *The Lancet Digital Health*, 2025.

[62] J. G. Verbalis, S. R. Goldsmith, A. Greenberg, C. Korzelius, R. W. Schrier, R. H. Sterns, and C. J. Thompson. Diagnosis, evaluation, and treatment of hyponatremia: expert panel recommendations. *The American journal of medicine*, 126(10):S1–S42, 2013.

[63] S. Vidros, C. Kolias, G. Kambourakis, and L. Akoglu. Automatic detection of online recruitment frauds: Characteristics, methods, and a public dataset. *Future Internet*, 9(1):6, 2017.

[64] P. Wan, Z. Huang, W. Tang, Y. Nie, D. Pei, S. Deng, J. Chen, Y. Zhou, H. Duan, Q. Chen, et al. Outpatient reception via collaboration between nurses and a large language model: a randomized controlled trial. *Nature Medicine*, 30(10):2878–2885, 2024.

[65] S. Waxler, P. Blazek, D. White, D. Sneider, K. Chung, M. Nagarathnam, P. Williams, H. Voeller, K. Wong, M. Swanhorst, et al. Generative medical event models improve with scale. *arXiv preprint arXiv:2508.12104*, 2025.

[66] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 24824–24837, 2022.

[67] P. K. Whelton, R. M. Carey, W. S. Aronow, D. E. Casey, K. J. Collins, C. Dennison Himmelfarb, S. M. DePalma, S. Gidding, K. A. Jamerson, D. W. Jones, et al. 2017 acc/aha/aapa/abc/acpm/ags/apha/ash/aspc/nma/pcna guideline for the prevention, detection, evaluation, and management of high blood pressure in adults: a report of the american college of cardiology/american heart association task force on clinical practice guidelines. *Journal of the American College of Cardiology*, 71(19):e127–e248, 2018.

[68] M. Wornow, R. Thapa, E. Steinberg, J. Fries, and N. Shah. EHRSHOT: An ehr benchmark for few-shot evaluation of foundation models. *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks*, 36:67125–67137, 2023.

[69] J. Wu, K. Xie, B. Gu, N. Krüger, K. J. Lin, and J. Yang. Why chain of thought fails in clinical text understanding. *arXiv preprint arXiv:2509.21933*, 2025.

[70] A. Yalamanchili, B. Sengupta, J. Song, S. Lim, T. O. Thomas, B. B. Mittal, M. E. Abazeed, and P. T. Teo. Quality of large language model responses to radiation oncology patient care questions. *JAMA network open*, 7(4):e244630, 2024.

[71] Y. Yamada, R. T. Lange, C. Lu, S. Hu, C. Lu, J. Foerster, J. Clune, and D. Ha. The ai scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint arXiv:2504.08066*, 2025.

[72] K. Yoon, L. Mao, C. Chong, T. J. Schwedt, C.-C. Chiang, and J. Li. Paregta: An llm-based ehr data encoding approach to capture temporal information. *arXiv preprint arXiv:2602.19661*, 2026.

[73] M. Zayyan. Objective structured clinical examination: the assessment of choice. *Oman medical journal*, 26(4):219, 2011.

[74] E. Zhang, R. Goto, N. Sagan, J. Mutter, N. Phillips, A. Alizadeh, K. Lee, J. Blanchet, M. Pilanci, and R. Tibshirani. Llm-lasso: A robust framework for domain-informed feature selection and regularization. *arXiv preprint arXiv:2502.10648*, 2025.

[75] S. Zhang, J. Fan, M. Fan, G. Li, and X. Du. Deepanalyze: Agentic large language models for autonomous data science. *arXiv preprint arXiv:2510.16872*, 2025.

[76] S. Zhang, Q. Liu, G. Qin, T. Naumann, and H. Poon. Med-rlvr: Emerging medical reasoning from a 3b base model via reinforcement learning. *arXiv preprint arXiv:2502.19655*, 2025.

[77] Y. Zhang, M. Li, D. Long, X. Zhang, H. Lin, B. Yang, P. Xie, A. Yang, D. Liu, J. Lin, F. Huang, and J. Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

A Extended Related Work

Data Science Agents. A complementary line of work studies "data science (DS) agents". Data Interpreter targets end-to-end problem solving, using code generation, execution, and revision to complete data analysis and mathematical tasks [16]. DS-Agent focuses on automating model development workflows such as task understanding, model selection, and training [13]. DeepAnalyze and DS-STAR push further toward autonomous data science over heterogeneous files, with an emphasis on multi-step data wrangling, open-ended querying, code execution, and report generation [37, 75]. This literature is closely related to our work in using LLMs as an interface to heterogeneous data. Our focus, however, is narrower and more controlled. Rather than asking LLM agents to plan and execute broad analyses, we study how LLMs can support input representation design with complex data for specific downstream tasks. This lets us isolate the role of representation choice and demonstrate its effect as a first-order driver of downstream statistical performance.

Medical QA Benchmarks. Recent work demonstrated that LLMs encode substantial clinical knowledge. Singhal et al. [52] introduced Med-PaLM, a closed-source medical LLM from Google. They evaluated it on MultiMedQA, a benchmark combining six medical question answering datasets, and showed that instruction-tuned models could surpass prior state-of-the-art. The follow-up Med-PaLM 2 pushed further, achieving performance competitive with expert physicians [53]. Sandmann et al. [47] evaluated open-source DeepSeek models on clinical decision support tasks using 125 patient cases. They found that the open-source frontier models perform equally well, if not better, than proprietary models. Zhang et al. [76] showed that even a 3B-parameter model can develop some medical reasoning skills when trained via reinforcement learning with verifiable rewards (RLVR, [27]) on the MedQA benchmark [21].

Moving Beyond Medical QA Benchmarks. While these results are impressive, high scores on medical QA do not translate to clinical impact. Mehandru et al. [35] proposed AI-based Standardized Clinical Examination (AI-SCE), modeled after the Objective Structured Clinical Examination (OSCE, [73]) used in medical training, to evaluate LLMs as agents in realistic, multi-step clinical scenarios rather than static question-answering benchmarks. Kim et al. [24] identified that LLMs can exhibit inflexible reasoning in clinical problem-solving and struggle with unexpected *long-tail* situations. Bedi et al. [4] introduced MedHELM, a holistic evaluation framework that organizes medical tasks into a taxonomy spanning clinical decision support, note generation, patient communication, research, and administration. Their analyses revealed that LLMs perform more variably on realistic clinical tasks than on standardized exams. Jiang et al. [20] developed MedAgentBench, a virtual EHR environment specifically designed to benchmark LLM agents on multi-step clinical tasks.

LLMs in the Clinic. Recent work probes how LLMs can be deployed to support clinicians and patients. Garcia et al. [10] demonstrated that AI-generated draft replies to patient messages can reduce physician burden without sacrificing quality. Yalamanchili et al. [70] evaluated the quality of LLM responses to radiation oncology patient queries. Unlu et al. [58] showed that retrieval-augmented GPT-4 (RAG, [29]) can assist with clinical trial screening. Randomized trials have also begun to probe whether LLMs can reliably improve clinicians' performance. Goh et al. [11] conducted a trial and found that GPT-4 assistance improved physician performance, while Wan et al. [64] showed that nurse-LLM collaboration for outpatient reception resulted in "increased satisfaction among both patients and nurses".

Conversational Diagnostic Models and Specialist Systems. Google's Articulate Medical Intelligence Explorer (AMIE) represents a sustained research program on LLMs in healthcare. Tu et al. [57] introduced AMIE, a conversational and diagnostic AI system, trained via self-play in a simulated environment to conduct multi-turn clinical conversations. They showed that it outperformed primary care physicians on 30 of 32 evaluation axes in a randomized, blinded study using standardized patient actors. McDuff et al. [34] evaluated AMIE's capacity for differential diagnosis, demonstrating that it generates diagnostic lists that exceed GPT-4's quality and improve clinicians' diagnostic accuracy when used as an assistive tool. Subsequent work extended AMIE to specialist domains. Palepu et al. [41] evaluated its performance in oncology care, and O'Sullivan et al. [40] reported results for complex cardiology cases.

LLMs with EHR Data. There is a growing literature developing medical foundation models pretrained on large EHR or claims data for risk prediction and trajectory modeling [46, 56, 65]. Recent work shows general-purpose LLMs can match domain-specific models on clinical tasks [15], which we reproduce and strengthen with our LLM-derived rubrics.

Agrawal et al. [2] showed that LLMs are effective clinical information extractors, pulling structured data from unstructured clinical text with few examples. Fleming et al. [9] released MedAlign, a clinician-generated dataset for instruction following, targeting realistic EHR-grounded tasks. Shi et al. [50] proposed EHRAgent, an LLM agent that generates and executes code to answer clinician queries. Lin et al. [31] combined supervised fine-tuning with reinforcement learning, demonstrating gains across medical calculation, patient-trial matching, and disease diagnosis in EHRSHOT benchmark. Liao et al. [30] developed EHR-R1, a reasoning-enhanced model for EHR analysis using reinforcement learning. Kirchler et al. [26] demonstrated that LLM-based clinical prediction models can have improved cross-country and system transferability. Yoon et al. [72] proposed an encoding approach for EHR data using LLMs to better emphasize temporal information.

Connections to AI for Science and Research. There is a growing literature on using LLMs to automate parts of the scientific process. Systems such as the AI Scientist [32, 71] generate hypotheses and iteratively design experiments. Other work emphasizes execution-grounded research pipelines, where LLM-generated plans are validated through execution and feedback [51]. Shao et al. [48] explore evolving rubrics to guide multi-step research. These directions are conceptually related to our setting: LLMs are used to generate intermediate representations and pipelines for complex tasks and datasets, that are subsequently executed and evaluated for downstream objectives.

B EHRSHOT Subsampling Procedure and Dataset Statistics

We evaluate on the 15 binary prediction tasks from EHRSHOT [68]. The tasks span four groups: operational outcomes, assignment of new diagnoses, anticipating laboratory results, and chest X-ray findings. Operational outcome tasks are visit-level predictions about near-term hospital events. Diagnosis tasks ask whether a patient will receive a new diagnosis within one year of the prediction time. Laboratory tasks ask whether the next relevant laboratory result will be abnormal. The chest X-ray task asks whether the associated radiology finding is abnormal.

B.1 Clinical Prediction Tasks.

We recall below the 15 binary prediction tasks across 4 taks categories:

- *Operational outcomes*:
  - ICU transfer: Will the patient be transferred to the intensive care unit?
  - Long length of stay: Will the patient stay in the hospital for more than 7 days?
  - 30-day readmission: Will the patient be readmitted to the hospital within 30 days?
- *Assignment of new diagnoses*:
  - Acute myocardial infarction (MI): Will the patient develop acute MI in the next year?
  - Celiac: Will the patient develop celiac in the next year?
  - Hyperlipidemia: Will the patient develop hyperlipidemia in the next year?
  - Hypertension: Will the patient develop hypertension in the next year?
  - Lupus: Will the patient develop lupus in the next year?
  - Pancreatic cancer: Will the patient develop pancreatic cancer in the next year?
- *Anticipating labs*:
  - Anemia: Will the patient's anemia lab come back as abnormal?
  - Hyperkalemia: Will the patient's hyperkalemia lab come back as abnormal?
  - Hypoglycemia: Will the patient's hypoglycemia lab come back as abnormal?
  - Hyponatremia: Will the patient's hyponatremia lab come back as abnormal?
  - Thrombocytopenia: Will the patient's thrombocytopenia lab come back as abnormal?
- *Chest X-ray*:
  - Abnormal chest x-ray findings: Will the patient's chest X-ray come back as abnormal?

B.2 Subsampling Procedure

The original EHRSHOT splits are used as the starting point for all tasks. Since several of our representation-learning methods require LLM calls per example, we subsample some splits to make the evaluation computationally feasible while preserving label balance where possible. The final sample counts used in all main experiments are reported in Table 1.

Let n_+ denote the number of positive examples in the corresponding original split. For validation sets, we construct label-balanced subsets for all tasks by sampling up to $\min(50, n_+)$ positive examples and the same number of negative examples. Thus, most validation sets contain 100 examples with 50 positives and 50 negatives; smaller validation sets occur only when fewer than 50 positive examples are available.

For operational outcome and diagnosis tasks, we keep the original EHRSHOT training and test splits unchanged. These tasks are moderate in size, so retaining the original train/test splits allows evaluation on the benchmark distribution while still reducing validation-time tuning cost.

For anticipating-lab and chest X-ray tasks, the original splits are substantially larger. We therefore also subsample the training and test splits by sampling up to $\min(1000, n_+)$ positive examples and the same number of negative examples in each split. This yields balanced train/test subsets of up to 2,000 examples per task. When fewer than 1,000 positives are available in a split, the corresponding number of negatives is matched to the available positives. This is why, for example, some lab test splits contain fewer than 2,000 total examples.

This subsampling strategy has two practical goals. First, it keeps LLM-based representation construction tractable across all tasks and methods. Second, it ensures that validation sets, and the subsampled lab/imaging splits, contain enough positive cases for stable model selection and evaluation.

Table 1: Number of samples (positive cases) per task and split. All splits for Lab results and Chest X-ray tasks are subsampled from the original dataset. Validation splits are subsampled for all tasks.

Category	Task	Train	Val	Test
Operational Outcomes (3)	ICU transfer	2402 (113)	100 (50)	2037 (85)
	Length of stay >7 days	2569 (681)	100 (50)	2195 (552)
	30-day readmission	2608 (370)	100 (50)	2189 (260)
Assignment of New Diagnosis (6)	Hypertension	1259 (182)	100 (50)	1258 (159)
	Hyperlipidemia	1684 (205)	100 (50)	1317 (172)
	Pancreatic cancer	2576 (155)	100 (50)	2220 (56)
	Celiac disease	2623 (62)	22 (11)	2222 (21)
	Lupus	2570 (104)	66 (33)	2243 (20)
	Acute MI	2534 (175)	100 (50)	2127 (144)
Anticipating Labs (5)	Thrombocytopenia	2000 (1000)	100 (50)	2000 (1000)
	Hyperkalemia	2000 (1000)	100 (50)	1896 (948)
	Hypoglycemia	2000 (1000)	100 (50)	1566 (783)
	Hyponatremia	2000 (1000)	100 (50)	2000 (1000)
	Anemia	2000 (1000)	100 (50)	2000 (1000)
Chest X-ray Findings (1)	Chest X-ray abnormality	2000 (1000)	100 (50)	2000 (1000)

C Additional Quantitative Results

C.1 Average Numerical Results for $n = 10$ and $n =$ All Regimes

Table 2: AUROC and AUPRC across all methods for $n = 10$ and $n =$ All. **Green**: best in entire column. **Blue**: best within a method across backbones. **Oper.:** Operational Outcomes, **New Dx:** Assignment of New Diagnosis, **Labs:** Anticipating Lab Results, **CXR:** Chest X-ray Findings

(a) AUROC

		Overall (15)	Oper. (3)	New Dx (6)	Labs (5)	CXR (1)
$n = 0$	Qwen3-8B-Zeroshot	.610$_{.601-.618}$	.639$_{.616-.660}$	.557$_{.541-.575}$	.667$_{.658-.677}$	.546$_{.519-.573}$
	GPT5-Mini-Zeroshot	.644$_{.635-.653}$	.680$_{.658-.700}$	.613$_{.594-.632}$	.684$_{.675-.694}$	.520$_{.498-.541}$
$n = 10$	Count-GBM	.594$_{.579-.609}$	.635$_{.610-.658}$	.619$_{.583-.652}$	.544$_{.532-.555}$	.571$_{.541-.602}$
	CLMBR-T	.597$_{.582-.613}$	.687$_{.662-.709}$	.593$_{.559-.629}$	.578$_{.567-.588}$	.453$_{.421-.484}$
	NaiveText	.609$_{.595-.621}$	.684$_{.664-.705}$	.621$_{.591-.649}$	.559$_{.547-.570}$	.558$_{.533-.584}$
	Local-Rubric	.701$_{.687-.715}$	.750$_{.730-.768}$	.713$_{.681-.743}$	.694$_{.684-.705}$	.517$_{.492-.543}$
	Local-Rubric-Basic	.662$_{.648-.675}$	.691$_{.670-.711}$	.711$_{.682-.741}$	.622$_{.610-.632}$	.487$_{.457-.519}$
	Local-Rubric-NoInterp	.687$_{.673-.701}$	.728$_{.708-.746}$	.711$_{.680-.741}$	.666$_{.656-.677}$	.525$_{.491-.556}$
	Global-Rubric-Blind	.632$_{.619-.646}$	.671$_{.651-.691}$	.643$_{.614-.672}$	.636$_{.624-.647}$	.437$_{.413-.461}$
	Global-Rubric	.649$_{.635-.662}$	.710$_{.690-.732}$	.657$_{.627-.688}$	.630$_{.619-.641}$	.513$_{.489-.538}$
	Global-Rubric-Auto	.652$_{.639-.666}$	.716$_{.695-.736}$	.653$_{.625-.683}$	.641$_{.630-.653}$	.503$_{.470-.534}$
	Global-Rubric-Tabular	.616$_{.605-.627}$	.601$_{.575-.628}$	.644$_{.619-.667}$	.614$_{.603-.625}$	.500$_{.500-.500}$
$n =$ All	Count-GBM	.685$_{.670-.699}$	.740$_{.717-.761}$	.755$_{.722-.786}$	.583$_{.572-.594}$	.609$_{.576-.642}$
	CLMBR-T	.725$_{.711-.739}$	.818$_{.799-.836}$	.697$_{.666-.728}$	.727$_{.717-.737}$	.609$_{.578-.640}$
	NaiveText	.699$_{.684-.714}$	.775$_{.754-.793}$	.709$_{.674-.744}$	.657$_{.646-.668}$	.616$_{.592-.640}$
	Local-Rubric	.772$_{.758-.784}$	.802$_{.786-.818}$	.770$_{.738-.799}$	.789$_{.780-.798}$	.606$_{.583-.630}$
	Local-Rubric-Basic	.737$_{.723-.752}$	.783$_{.765-.800}$	.725$_{.692-.758}$	.752$_{.742-.761}$	.599$_{.568-.628}$
	Local-Rubric-NoInterp	.770$_{.757-.782}$	.797$_{.779-.815}$	.776$_{.747-.803}$	.780$_{.771-.789}$	.596$_{.565-.625}$
	Global-Rubric-Blind	.751$_{.738-.764}$	.776$_{.755-.794}$	.745$_{.716-.775}$	.777$_{.768-.787}$	.585$_{.559-.608}$
	Global-Rubric	.763$_{.748-.777}$	.786$_{.768-.805}$	.756$_{.723-.789}$	.791$_{.781-.800}$	.594$_{.567-.619}$
	Global-Rubric-Auto	.756$_{.743-.769}$	.790$_{.772-.807}$	.752$_{.722-.782}$	.776$_{.766-.786}$	.575$_{.543-.605}$
	Global-Rubric-Tabular	.752$_{.740-.764}$	.766$_{.748-.785}$	.737$_{.710-.763}$	.804$_{.794-.813}$	.540$_{.507-.572}$

(b) AUPRC

		Overall (15)	Oper. (3)	New Dx (6)	Labs (5)	CXR (1)
$n = 0$	Qwen3-8B-Zeroshot	.316$_{.309-.324}$	.185$_{.172-.200}$	.080$_{.069-.093}$	.637$_{.623-.650}$	.526$_{.491-.560}$
	GPT5-Mini-Zeroshot	.348$_{.337-.362}$	.221$_{.205-.240}$	.137$_{.111-.169}$	.647$_{.634-.659}$	.507$_{.476-.538}$
$n = 10$	Count-GBM	.298$_{.289-.307}$	.213$_{.193-.235}$	.098$_{.084-.116}$	.540$_{.526-.555}$	.543$_{.502-.583}$
	CLMBR-T	.315$_{.305-.326}$	.265$_{.237-.294}$	.099$_{.084-.116}$	.575$_{.561-.590}$	.465$_{.427-.500}$
	NaiveText	.306$_{.296-.316}$	.225$_{.200-.253}$	.097$_{.081-.114}$	.557$_{.541-.572}$	.551$_{.519-.583}$
	Local-Rubric	.382$_{.369-.396}$	.270$_{.244-.296}$	.170$_{.143-.198}$	.680$_{.664-.694}$	.503$_{.474-.535}$
	Local-Rubric-Basic	.345$_{.333-.359}$	.219$_{.199-.240}$	.163$_{.136-.193}$	.610$_{.595-.624}$	.493$_{.454-.531}$
	Local-Rubric-NoInterp	.371$_{.356-.387}$	.257$_{.232-.284}$	.175$_{.144-.206}$	.649$_{.634-.663}$	.509$_{.469-.546}$
	Global-Rubric-Blind	.333$_{.322-.345}$	.207$_{.189-.230}$	.133$_{.112-.155}$	.625$_{.609-.639}$	.455$_{.427-.486}$
	Global-Rubric	.345$_{.332-.359}$	.241$_{.219-.264}$	.147$_{.120-.177}$	.610$_{.595-.625}$	.521$_{.489-.552}$
	Global-Rubric-Auto	.341$_{.331-.353}$	.254$_{.229-.283}$	.120$_{.104-.139}$	.626$_{.610-.642}$	.497$_{.460-.534}$
	Global-Rubric-Tabular	.303$_{.296-.310}$	.184$_{.169-.200}$	.091$_{.081-.102}$	.589$_{.576-.603}$	.497$_{.470-.523}$
$n =$ All	Count-GBM	.371$_{.353-.388}$	.274$_{.250-.302}$	.218$_{.180-.255}$	.570$_{.556-.584}$	.582$_{.540-.624}$
	CLMBR-T	.430$_{.417-.444}$	.425$_{.387-.468}$	.170$_{.145-.197}$	.713$_{.699-.726}$	.600$_{.559-.643}$
	NaiveText	.391$_{.377-.406}$	.315$_{.283-.346}$	.179$_{.151-.208}$	.649$_{.634-.664}$	.609$_{.577-.641}$
	Local-Rubric	.452$_{.439-.466}$	.341$_{.310-.374}$	.223$_{.194-.251}$	.762$_{.748-.776}$	.605$_{.571-.638}$
	Local-Rubric-Basic	.431$_{.417-.445}$	.333$_{.304-.368}$	.203$_{.174-.234}$	.731$_{.716-.744}$	.594$_{.554-.633}$
	Local-Rubric-NoInterp	.451$_{.437-.465}$	.345$_{.315-.379}$	.227$_{.198-.257}$	.758$_{.744-.771}$	.581$_{.537-.625}$
	Global-Rubric-Blind	.428$_{.415-.443}$	.321$_{.292-.351}$	.185$_{.155-.217}$	.753$_{.739-.767}$	.587$_{.556-.620}$
	Global-Rubric	.459$_{.442-.478}$	.339$_{.309-.371}$	.236$_{.200-.276}$	.773$_{.760-.786}$	.582$_{.550-.614}$
	Global-Rubric-Auto	.437$_{.422-.452}$	.343$_{.312-.378}$	.195$_{.164-.226}$	.760$_{.745-.773}$	.557$_{.518-.596}$
	Global-Rubric-Tabular	.445$_{.429-.461}$	.329$_{.296-.363}$	.208$_{.175-.243}$	.783$_{.769-.796}$	.527$_{.486-.567}$

C.2 Sample-size Sweep Figures with Confidence Bands

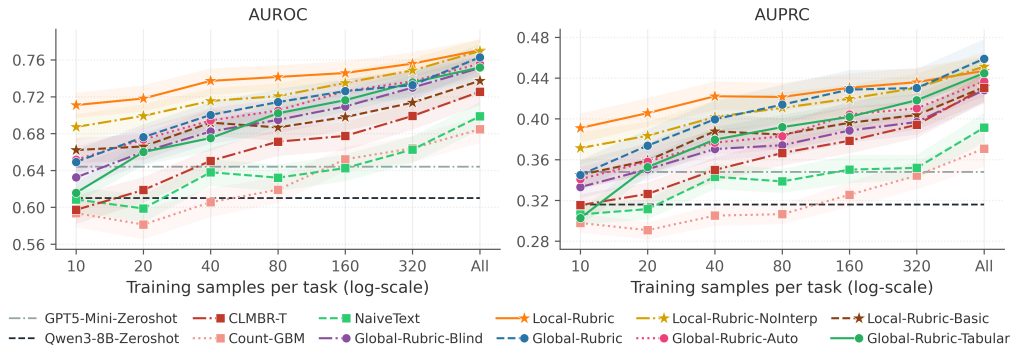


Figure 8: Average overall performance across different sample sizes. Bands denote 95% CI, bootstrapped over the test set.

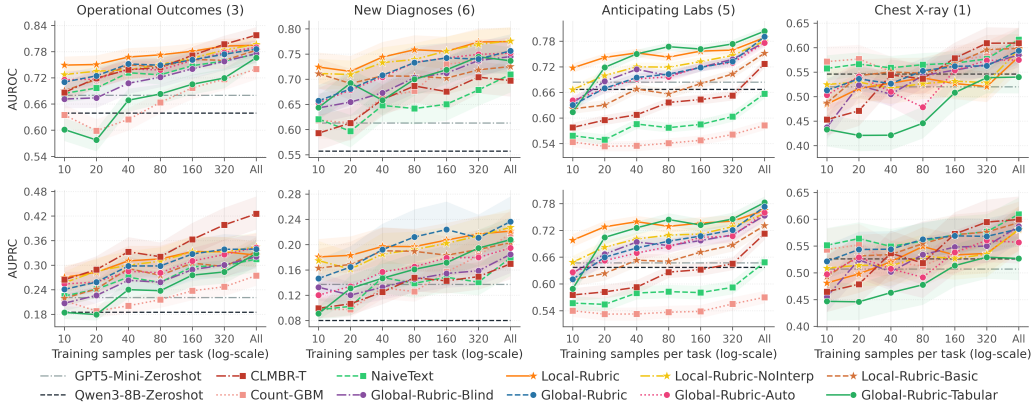


Figure 9: Average performance for each task type across different sample sizes. Bands denote 95% CI, bootstrapped over the test set.

C.3 Full EHRSHOT Evaluation Results for Global-Rubric-Tabular

Once a global rubric and its associated tabularization script have been learned, applying the transformation to additional examples is inexpensive and deterministic. This makes it practical to evaluate Global-Rubric-Tabular on the full EHRSHOT dataset without subsampling. Table 3 reports results for all 15 benchmark tasks. The model achieves a mean AUROC of 0.770 and mean AUPRC of 0.312 across tasks.

Table 3: Full-dataset EHRSHOT results for Global-Rubric-Tabular.

Task	AUROC	AUPRC
Operational Outcomes		
ICU transfer	$.800_{.752–.846}$	$.195_{.130–.277}$
Long length of stay	$.737_{.714–.758}$	$.464_{.421–.509}$
30-day readmit	$.757_{.725–.787}$	$.357_{.300–.419}$
Group Avg.	$\mathbf{.765}_{.745–.786}$	$\mathbf{.340}_{.308–.374}$
Assignment of New Diagnoses		
Acute MI	$.765_{.726–.803}$	$.183_{.140–.225}$
Lupus	$.819_{.728–.897}$	$.048_{.020–.078}$
Hyperlipidemia	$.736_{.697–.772}$	$.312_{.248–.383}$
Hypertension	$.718_{.677–.757}$	$.299_{.239–.368}$
Celiac disease	$.654_{.540–.772}$	$.098_{.016–.190}$
Pancreatic cancer	$.859_{.798–.915}$	$.378_{.239–.518}$
Group Avg.	$\mathbf{.759}_{.729–.790}$	$\mathbf{.216}_{.185–.248}$
Anticipating Lab Results		
Anemia	$.810_{.806–.815}$	$.361_{.351–.372}$
Hyponatremia	$.815_{.811–.818}$	$.560_{.552–.568}$
Thrombocytopenia	$.885_{.881–.888}$	$.571_{.558–.582}$
Hyperkalemia	$.853_{.840–.865}$	$.097_{.086–.109}$
Hypoglycemia	$.751_{.733–.768}$	$.031_{.026–.037}$
Group Avg.	$\mathbf{.823}_{.818–.827}$	$\mathbf{.324}_{.319–.329}$
Chest X-ray Findings		
Chest X-ray	$.584_{.572–.597}$	$.743_{.731–.756}$
Group Avg.	$\mathbf{.584}_{.571–.597}$	$\mathbf{.743}_{.731–.756}$
Overall Avg. (15 tasks)	$\mathbf{.770}_{.756–.783}$	$\mathbf{.312}_{.297–.327}$

C.4 Token Counts of Different Textual Representations

Table 4: Token count statistics for the textual representations that are passed into the text embedding model prior to downstream learning across different methods. Statistics for each representation are computed after merging data from all 15 tasks and 3 splits in Table 1. Qwen3-8B tokenizer is used.

Representation	Mean	Median	Min	Max	Std
NaiveText	5,980	7,769	131	8,267	2,648
Local-Rubric	853	844	492	1,430	116
Global-Rubric	1,283	1,193	318	3,136	452

C.5 Results Using a Different Text-embedding Models as Backbones in Downstream Training

Table 5: Performance with different text-embedding models (n = All). **Green**: best in entire column. **Blue**: best within a method across backbones. **Oper.:** Operational Outcomes, **New Dx:** Assignment of New Diagnosis, **Labs:** Anticipating Lab Results, **CXR:** Chest X-ray Findings.

(A) AUROC

Method	Backbone	Overall	Oper. (3)	New Dx (6)	Labs (5)	CXR (1)
NaiveText	Mistral-7B	.612$_{.596-.626}$	.656$_{.631-.681}$	.647$_{.610-.681}$	.574$_{.562-.585}$	.453$_{.421-.483}$
	Llama3-8B	.594$_{.578-.609}$	.660$_{.633-.687}$	.619$_{.584-.654}$	.552$_{.540-.563}$	.463$_{.433-.494}$
	TE3-L	.674$_{.660-.689}$	.741$_{.719-.761}$	.713$_{.681-.744}$	.610$_{.599-.621}$	.564$_{.532-.596}$
	Qwen3-8B	.699$_{.684-.714}$	.775$_{.754-.793}$	.709$_{.674-.744}$	.657$_{.646-.668}$	.616$_{.592-.640}$
Local-Rubric	Mistral-7B	.760$_{.746-.773}$	.804$_{.787-.820}$	.762$_{.727-.794}$	.768$_{.759-.778}$	.569$_{.537-.598}$
	Llama3-8B	.761$_{.746-.775}$	.793$_{.774-.810}$	.764$_{.726-.796}$	.772$_{.762-.780}$	.591$_{.560-.621}$
	TE3-L	.762$_{.749-.774}$	.798$_{.781-.815}$	.762$_{.732-.790}$	.782$_{.773-.790}$	.553$_{.523-.585}$
	Qwen3-8B	.772$_{.758-.784}$	.802$_{.786-.818}$	.770$_{.738-.799}$	.789$_{.780-.798}$	.606$_{.583-.630}$
Global-Rubric	Mistral-7B	.725$_{.711-.739}$	.762$_{.742-.782}$	.692$_{.659-.724}$	.775$_{.766-.785}$	.557$_{.532-.583}$
	Llama3-8B	.736$_{.722-.750}$	.764$_{.744-.784}$	.715$_{.684-.747}$	.777$_{.768-.787}$	.565$_{.540-.590}$
	TE3-L	.738$_{.726-.750}$	.764$_{.743-.784}$	.714$_{.687-.742}$	.788$_{.778-.796}$	.561$_{.537-.584}$
	Qwen3-8B	.763$_{.748-.777}$	.786$_{.768-.805}$	.756$_{.723-.789}$	.791$_{.781-.800}$	.594$_{.567-.619}$

(B) AUPRC

Method	Backbone	Overall	Oper. (3)	New Dx (6)	Labs (5)	CXR (1)
NaiveText	Mistral-7B	.316$_{.304-.328}$	.244$_{.220-.274}$	.124$_{.103-.147}$	.561$_{.546-.576}$	.457$_{.424-.489}$
	Llama3-8B	.310$_{.301-.321}$	.241$_{.218-.269}$	.116$_{.099-.133}$	.549$_{.534-.563}$	.494$_{.455-.532}$
	TE3-L	.351$_{.340-.364}$	.282$_{.254-.311}$	.145$_{.122-.173}$	.601$_{.586-.615}$	.546$_{.508-.584}$
	Qwen3-8B	.391$_{.377-.406}$	.315$_{.283-.346}$	.179$_{.151-.208}$	.649$_{.634-.664}$	.609$_{.577-.641}$
Local-Rubric	Mistral-7B	.446$_{.430-.463}$	.357$_{.326-.389}$	.224$_{.188-.259}$	.741$_{.727-.754}$	.573$_{.533-.613}$
	Llama3-8B	.450$_{.433-.467}$	.342$_{.310-.376}$	.235$_{.199-.270}$	.745$_{.731-.759}$	.584$_{.546-.624}$
	TE3-L	.445$_{.431-.461}$	.348$_{.315-.380}$	.218$_{.187-.252}$	.755$_{.741-.768}$	.559$_{.518-.599}$
	Qwen3-8B	.452$_{.439-.466}$	.341$_{.310-.374}$	.223$_{.194-.251}$	.762$_{.748-.776}$	.605$_{.571-.638}$
Global-Rubric	Mistral-7B	.417$_{.404-.430}$	.338$_{.307-.370}$	.153$_{.129-.178}$	.755$_{.742-.767}$	.548$_{.516-.579}$
	Llama3-8B	.430$_{.417-.445}$	.337$_{.307-.368}$	.179$_{.153-.208}$	.760$_{.748-.773}$	.565$_{.533-.598}$
	TE3-L	.433$_{.419-.447}$	.346$_{.311-.380}$	.178$_{.153-.205}$	.771$_{.758-.784}$	.531$_{.502-.563}$
	Qwen3-8B	.459$_{.442-.478}$	.339$_{.309-.371}$	.236$_{.200-.276}$	.773$_{.760-.786}$	.582$_{.550-.614}$

We evaluate four text embedding models for the downstream learning step, keeping the classifier (logistic regression on frozen embeddings) fixed.

We compare **Qwen3-Embedding-8B** [77], two **LLM2Vec** [5] bidirectional adaptations of **Llama-3-8B-Instruct** [36] and **Mistral-7B-Instruct** [19], and the proprietary **text-embedding-3-large** model (**TE3-L**) accessed through the OpenAI API [38]. We use n = All and compare the main textual representation-based methods: NaiveText, Local-Rubric, and Global-Rubric.

As shown in Table 5, Qwen3-Embedding-8B achieves the highest overall AUROC and AUPRC across methods compared. Within different task types, it continues to be either the best text-embedding model, or very close to top model. In the remaining parts of the paper, we report results for Qwen3-Embedding-8B model, unless otherwise stated.

C.6 Per-task results

In this section, we report AUROC and AUPRC metrics separately for each task in Figures 10–13 and Tables 6–20. Qwen3-Embedding-8B is used as the text-embedding model. For some results with other embedding models, see Appendix C.5.

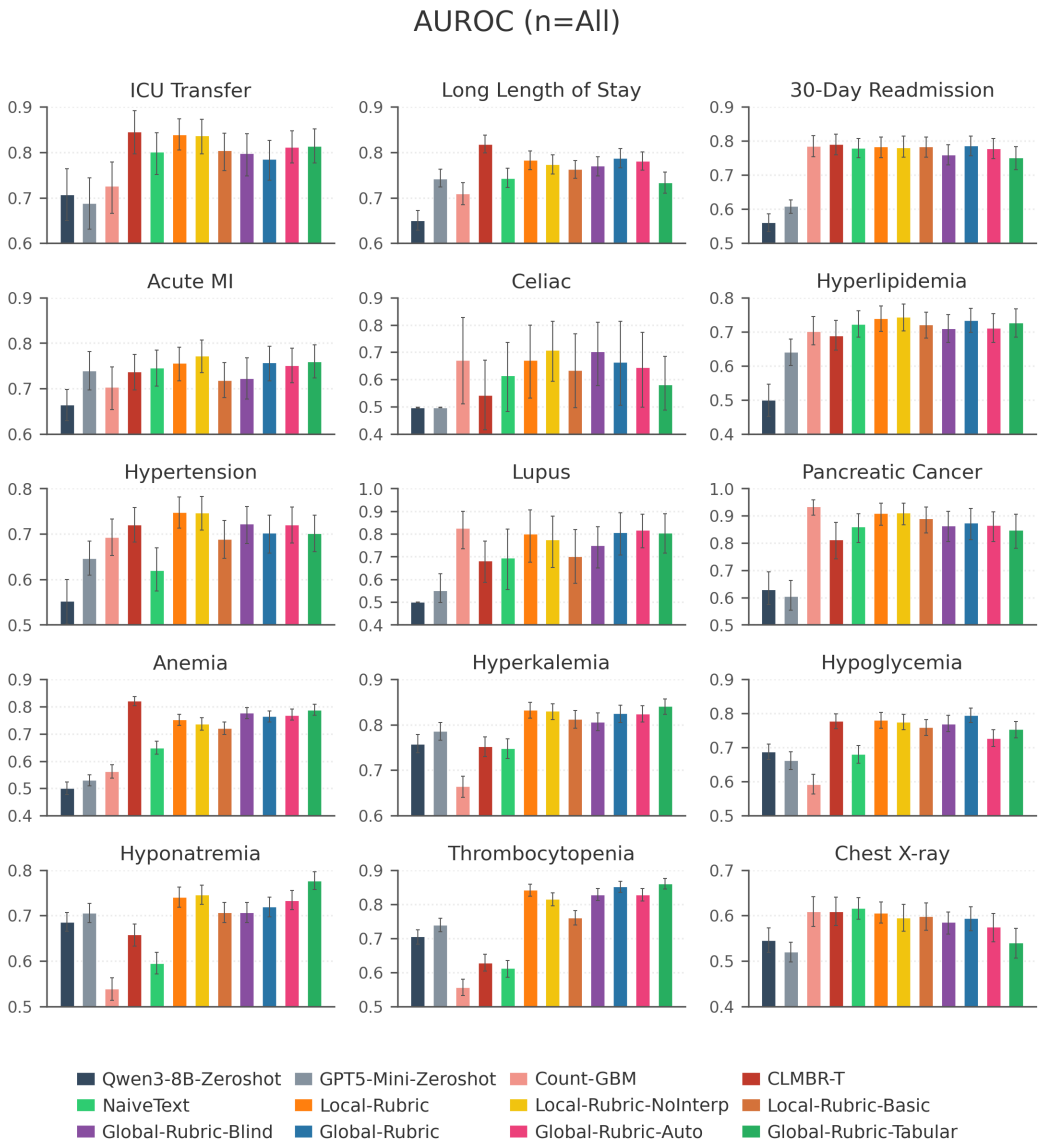


Figure 10: AUROC per-task, $n = \text{All}$

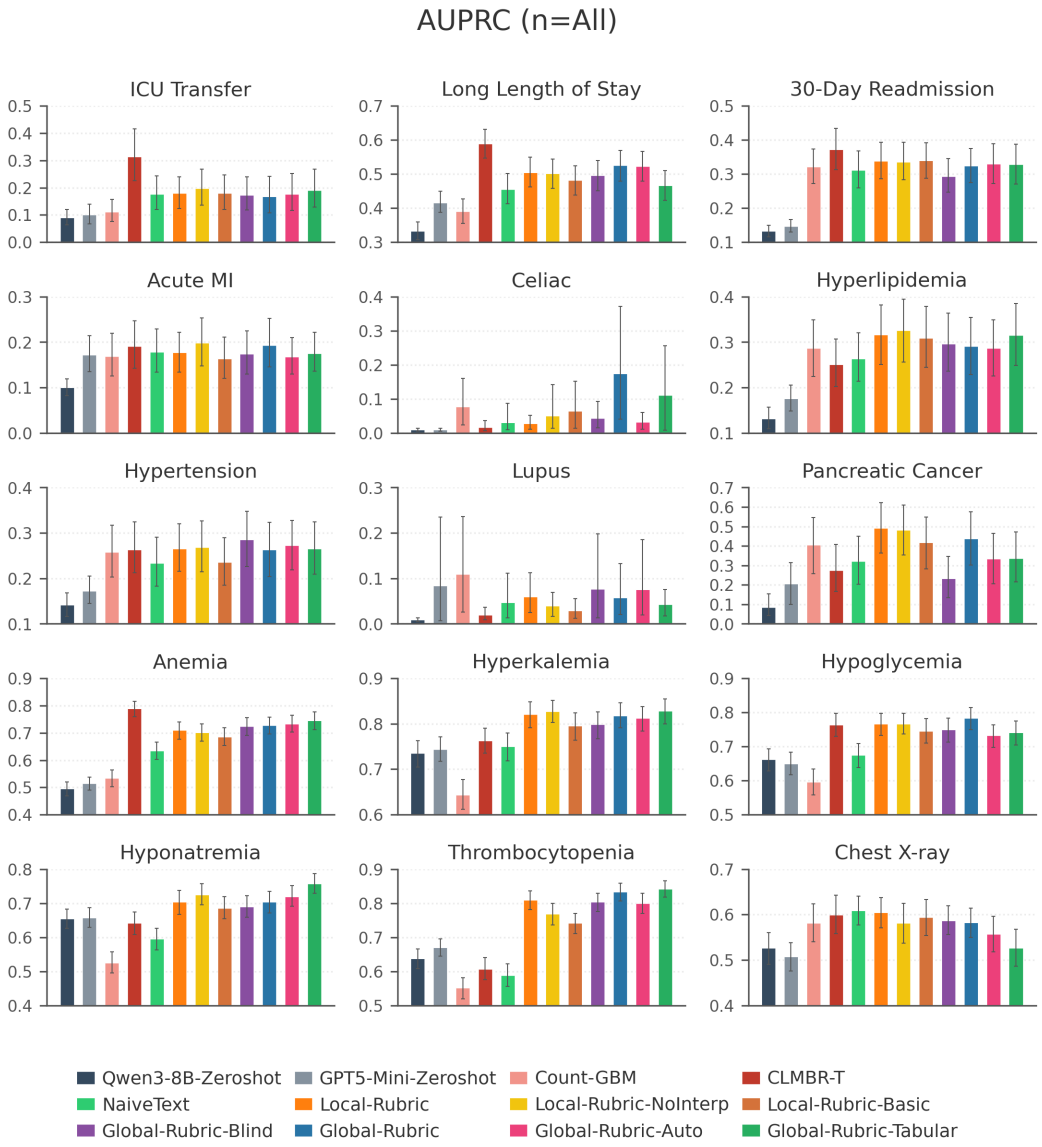


Figure 11: AUPRC per-task, n = All

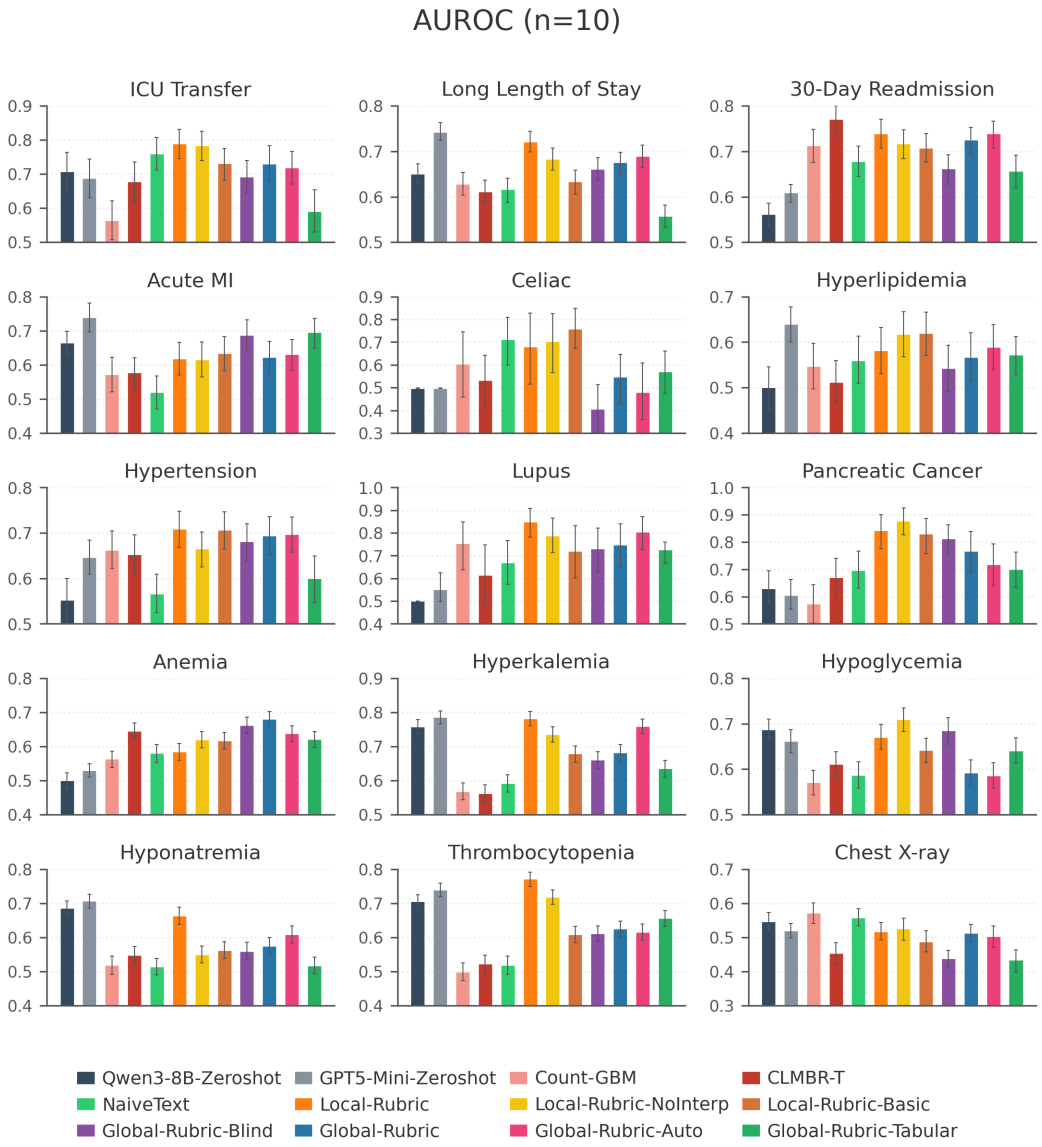


Figure 12: AUROC per-task, $n = 10$

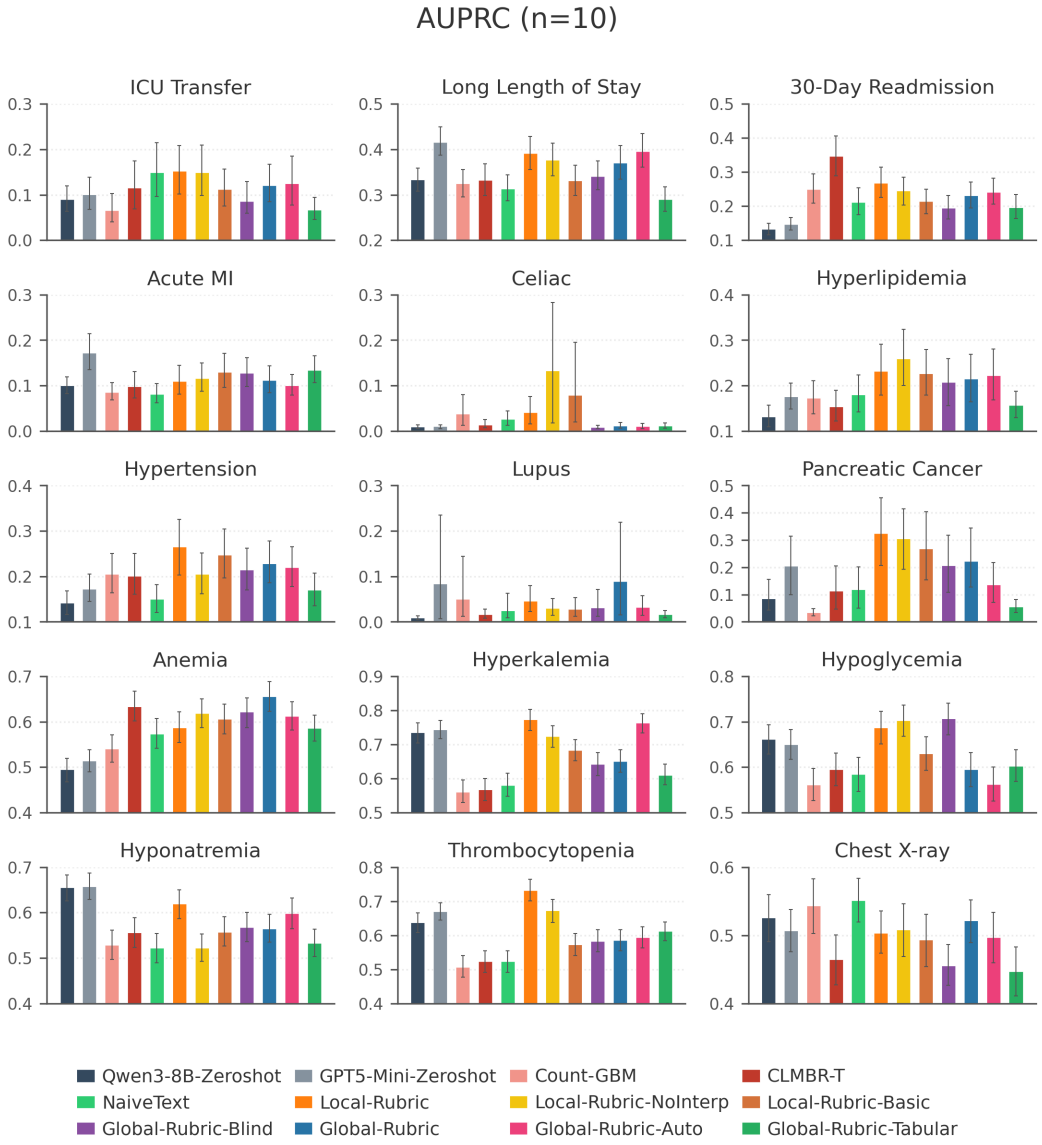


Figure 13: AUPRC per-task, $n = 10$

Table 6: Guo ICU. AUROC and AUPRC with 95% bootstrap CI. Best per column for each setting is highlighted. Embeddings are from Qwen3-8B; sample sizes refer to per-class training examples.

	AUROC		**AUPRC**	
Qwen3-8B-Zeroshot	$.707_{.650-.764}$		$.090_{.064-.120}$	
GPT5-Mini-Zeroshot	$.688_{.631-.744}$		$.101_{.068-.139}$	
	$n = 10$	$n = \text{All}$	$n = 10$	$n = \text{All}$
Count-GBM	$.564_{.508-.622}$	$.726_{.665-.779}$	$.066_{.040-.103}$	$.112_{.076-.157}$
CLMBR-T	$.678_{.617-.736}$	$.845_{.797-.892}$	$.115_{.069-.174}$	$.314_{.226-.416}$
NaiveText	$.759_{.711-.807}$	$.801_{.751-.843}$	$.149_{.097-.214}$	$.177_{.119-.243}$
Local-Rubric	$.789_{.745-.832}$	$.839_{.805-.873}$	$.152_{.102-.209}$	$.179_{.124-.239}$
Local-Rubric-Basic	$.731_{.681-.775}$	$.804_{.759-.842}$	$.113_{.075-.156}$	$.179_{.120-.247}$
Local-Rubric-NoInterp	$.783_{.739-.826}$	$.837_{.797-.873}$	$.150_{.098-.209}$	$.197_{.136-.267}$
Global-Rubric-Blind	$.692_{.642-.740}$	$.797_{.748-.841}$	$.086_{.059-.130}$	$.173_{.118-.240}$
Global-Rubric	$.730_{.680-.784}$	$.785_{.738-.827}$	$.121_{.085-.167}$	$.168_{.108-.242}$
Global-Rubric-Auto	$.719_{.671-.766}$	$.811_{.777-.847}$	$.125_{.077-.185}$	$.177_{.116-.252}$
Global-Rubric-Tabular	$.591_{.529-.653}$	$.814_{.777-.851}$	$.067_{.045-.095}$	$.191_{.128-.268}$

Table 7: Guo length of stay. AUROC and AUPRC with 95% bootstrap CI. Best per column for each setting is highlighted. Embeddings are from Qwen3-8B; sample sizes refer to per-class training examples.

	AUROC		**AUPRC**	
Qwen3-8B-Zeroshot	$.650_{.629-.672}$		$.333_{.308-.359}$	
GPT5-Mini-Zeroshot	$.742_{.724-.763}$		$.416_{.387-.449}$	
	$n = 10$	$n = \text{All}$	$n = 10$	$n = \text{All}$
Count-GBM	$.628_{.604-.654}$	$.709_{.685-.733}$	$.325_{.295-.355}$	$.390_{.355-.426}$
CLMBR-T	$.611_{.587-.637}$	$.818_{.799-.837}$	$.332_{.298-.368}$	$.589_{.546-.632}$
NaiveText	$.616_{.588-.641}$	$.743_{.723-.765}$	$.313_{.286-.344}$	$.455_{.413-.502}$
Local-Rubric	$.721_{.698-.744}$	$.783_{.762-.803}$	$.391_{.356-.428}$	$.505_{.462-.550}$
Local-Rubric-Basic	$.633_{.606-.659}$	$.763_{.743-.782}$	$.331_{.299-.365}$	$.482_{.438-.524}$
Local-Rubric-NoInterp	$.683_{.659-.707}$	$.773_{.752-.794}$	$.376_{.342-.414}$	$.502_{.458-.544}$
Global-Rubric-Blind	$.661_{.637-.686}$	$.770_{.748-.791}$	$.341_{.311-.374}$	$.496_{.450-.540}$
Global-Rubric	$.675_{.649-.698}$	$.787_{.766-.809}$	$.370_{.334-.408}$	$.526_{.480-.570}$
Global-Rubric-Auto	$.689_{.664-.714}$	$.781_{.761-.801}$	$.396_{.361-.434}$	$.523_{.479-.567}$
Global-Rubric-Tabular	$.557_{.533-.582}$	$.734_{.710-.757}$	$.290_{.264-.317}$	$.467_{.423-.510}$

Table 8: Guo readmission. AUROC and AUPRC with 95% bootstrap CI. Best per column for each setting is highlighted. Embeddings are from Qwen3-8B; sample sizes refer to per-class training examples.

	AUROC		**AUPRC**	
Qwen3-8B-Zeroshot	$.561_{.534-.586}$		$.133_{.116-.150}$	
GPT5-Mini-Zeroshot	$.609_{.588-.627}$		$.147_{.130-.166}$	
	$n = 10$	$n = \text{All}$	$n = 10$	$n = \text{All}$
Count-GBM	$.712_{.676-.748}$	$.785_{.754-.816}$	$.249_{.208-.295}$	$.321_{.272-.373}$
CLMBR-T	$.771_{.738-.800}$	$.791_{.760-.819}$	$.347_{.289-.405}$	$.373_{.313-.434}$
NaiveText	$.678_{.645-.711}$	$.780_{.751-.808}$	$.212_{.176-.253}$	$.312_{.259-.368}$
Local-Rubric	$.739_{.707-.771}$	$.783_{.752-.812}$	$.267_{.226-.315}$	$.339_{.286-.393}$
Local-Rubric-Basic	$.707_{.677-.739}$	$.783_{.752-.811}$	$.214_{.178-.250}$	$.339_{.288-.392}$
Local-Rubric-NoInterp	$.717_{.684-.747}$	$.781_{.752-.814}$	$.245_{.203-.285}$	$.335_{.283-.393}$
Global-Rubric-Blind	$.662_{.628-.693}$	$.760_{.729-.788}$	$.195_{.162-.231}$	$.293_{.246-.346}$
Global-Rubric	$.726_{.695-.753}$	$.786_{.757-.814}$	$.232_{.195-.271}$	$.324_{.275-.375}$
Global-Rubric-Auto	$.739_{.708-.766}$	$.778_{.748-.807}$	$.242_{.206-.282}$	$.329_{.272-.390}$
Global-Rubric-Tabular	$.656_{.619-.692}$	$.751_{.715-.783}$	$.196_{.163-.233}$	$.328_{.271-.387}$

Table 9: Acute MI. AUROC and AUPRC with 95% bootstrap CI. Best per column for each setting is highlighted. Embeddings are from Qwen3-8B; sample sizes refer to per-class training examples.

	AUROC		**AUPRC**	
Qwen3-8B-Zeroshot	$.664_{.629-.698}$		$.100_{.082-.119}$	
GPT5-Mini-Zeroshot	$.739_{.697-.782}$		$.172_{.135-.214}$	
	$n = 10$	$n = \text{All}$	$n = 10$	$n = \text{All}$
Count-GBM	$.572_{.520-.622}$	$.704_{.654-.748}$	$.086_{.068-.106}$	$.169_{.125-.219}$
CLMBR-T	$.577_{.528-.621}$	$.737_{.697-.775}$	$.099_{.073-.131}$	$.191_{.142-.247}$
NaiveText	$.520_{.470-.568}$	$.746_{.706-.785}$	$.081_{.062-.105}$	$.179_{.134-.229}$
Local-Rubric	$.618_{.570-.665}$	$.756_{.717-.791}$	$.110_{.081-.144}$	$.177_{.134-.222}$
Local-Rubric-Basic	$.633_{.583-.683}$	$.718_{.680-.757}$	$.129_{.096-.171}$	$.163_{.120-.211}$
Local-Rubric-NoInterp	$.616_{.564-.667}$	$.772_{.735-.807}$	$.116_{.088-.150}$	$.199_{.148-.253}$
Global-Rubric-Blind	$.687_{.643-.732}$	$.723_{.677-.768}$	$.128_{.098-.161}$	$.174_{.130-.225}$
Global-Rubric	$.623_{.572-.669}$	$.757_{.717-.793}$	$.112_{.084-.144}$	$.194_{.146-.253}$
Global-Rubric-Auto	$.631_{.585-.675}$	$.751_{.713-.789}$	$.100_{.079-.125}$	$.168_{.130-.211}$
Global-Rubric-Tabular	$.695_{.649-.737}$	$.760_{.723-.797}$	$.134_{.107-.165}$	$.176_{.136-.221}$

Table 10: Celiac disease. AUROC and AUPRC with 95% bootstrap CI. Best per column for each setting is highlighted. Embeddings are from Qwen3-8B; sample sizes refer to per-class training examples.

	AUROC		**AUPRC**	
Qwen3-8B-Zeroshot	$.497_{.495-.498}$		$.009_{.005-.014}$	
GPT5-Mini-Zeroshot	$.497_{.496-.499}$		$.010_{.005-.014}$	
	$n = 10$	$n = \text{All}$	$n = 10$	$n = \text{All}$
Count-GBM	$.603_{.457-.745}$	$.671_{.510-.828}$	$.038_{.012-.080}$	$.077_{.024-.160}$
CLMBR-T	$.532_{.409-.642}$	$.543_{.415-.671}$	$.013_{.007-.025}$	$.017_{.007-.036}$
NaiveText	$.712_{.599-.810}$	$.614_{.482-.736}$	$.026_{.012-.044}$	$.030_{.009-.087}$
Local-Rubric	$.680_{.515-.828}$	$.670_{.532-.799}$	$.041_{.016-.076}$	$.028_{.012-.052}$
Local-Rubric-Basic	$.757_{.671-.849}$	$.634_{.497-.768}$	$.079_{.020-.195}$	$.064_{.014-.152}$
Local-Rubric-NoInterp	$.704_{.567-.826}$	$.708_{.594-.813}$	$.132_{.018-.283}$	$.050_{.014-.141}$
Global-Rubric-Blind	$.406_{.304-.514}$	$.702_{.578-.810}$	$.009_{.005-.013}$	$.043_{.015-.092}$
Global-Rubric	$.546_{.428-.646}$	$.663_{.505-.814}$	$.012_{.007-.019}$	$.174_{.040-.372}$
Global-Rubric-Auto	$.480_{.359-.609}$	$.644_{.499-.774}$	$.011_{.006-.017}$	$.032_{.012-.060}$
Global-Rubric-Tabular	$.570_{.474-.660}$	$.582_{.487-.686}$	$.012_{.007-.017}$	$.111_{.008-.255}$

Table 11: Hyperlipidemia. AUROC and AUPRC with 95% bootstrap CI. Best per column for each setting is highlighted. Embeddings are from Qwen3-8B; sample sizes refer to per-class training examples.

	AUROC		**AUPRC**	
Qwen3-8B-Zeroshot	$.500_{.452-.546}$		$.132_{.109-.157}$	
GPT5-Mini-Zeroshot	$.640_{.601-.678}$		$.176_{.148-.205}$	
	$n = 10$	$n = \text{All}$	$n = 10$	$n = \text{All}$
Count-GBM	$.546_{.496-.597}$	$.702_{.662-.745}$	$.172_{.138-.211}$	$.287_{.225-.349}$
CLMBR-T	$.512_{.466-.560}$	$.689_{.647-.733}$	$.153_{.122-.189}$	$.251_{.202-.307}$
NaiveText	$.559_{.510-.613}$	$.722_{.684-.762}$	$.180_{.142-.223}$	$.263_{.213-.320}$
Local-Rubric	$.581_{.531-.632}$	$.740_{.701-.776}$	$.232_{.179-.291}$	$.316_{.251-.382}$
Local-Rubric-Basic	$.619_{.571-.666}$	$.721_{.681-.758}$	$.226_{.179-.280}$	$.309_{.245-.379}$
Local-Rubric-NoInterp	$.617_{.568-.667}$	$.743_{.703-.782}$	$.259_{.200-.323}$	$.326_{.256-.394}$
Global-Rubric-Blind	$.542_{.491-.593}$	$.710_{.668-.750}$	$.208_{.156-.259}$	$.297_{.236-.364}$
Global-Rubric	$.567_{.516-.620}$	$.734_{.698-.769}$	$.215_{.165-.269}$	$.291_{.228-.355}$
Global-Rubric-Auto	$.589_{.540-.639}$	$.711_{.669-.753}$	$.222_{.168-.280}$	$.286_{.225-.349}$
Global-Rubric-Tabular	$.572_{.529-.612}$	$.727_{.684-.768}$	$.157_{.129-.187}$	$.316_{.248-.385}$

Table 12: Hypertension. AUROC and AUPRC with 95% bootstrap CI. Best per column for each setting is highlighted. Embeddings are from Qwen3-8B; sample sizes refer to per-class training examples.

	AUROC		**AUPRC**	
Qwen3-8B-Zeroshot	$.552_{.502-.600}$		$.142_{.116-.168}$	
GPT5-Mini-Zeroshot	$.646_{.609-.684}$		$.172_{.145-.205}$	
	$n = 10$	$n = \text{All}$	$n = 10$	$n = \text{All}$
Count-GBM	$.662_{.622-.704}$	$.693_{.653-.732}$	$.206_{.164-.251}$	$.258_{.203-.317}$
CLMBR-T	$.653_{.611-.696}$	$.721_{.682-.759}$	$.201_{.160-.251}$	$.263_{.213-.324}$
NaiveText	$.566_{.525-.609}$	$.620_{.574-.669}$	$.150_{.121-.182}$	$.234_{.183-.291}$
Local-Rubric	$.709_{.669-.747}$	$.747_{.713-.782}$	$.265_{.203-.325}$	$.265_{.215-.320}$
Local-Rubric-Basic	$.707_{.664-.747}$	$.688_{.646-.730}$	$.247_{.196-.305}$	$.236_{.185-.290}$
Local-Rubric-NoInterp	$.665_{.625-.702}$	$.746_{.709-.782}$	$.205_{.162-.251}$	$.268_{.215-.326}$
Global-Rubric-Blind	$.681_{.637-.721}$	$.722_{.679-.761}$	$.214_{.170-.262}$	$.285_{.226-.347}$
Global-Rubric	$.693_{.653-.736}$	$.702_{.658-.742}$	$.228_{.186-.278}$	$.263_{.204-.323}$
Global-Rubric-Auto	$.696_{.658-.735}$	$.721_{.680-.759}$	$.220_{.178-.265}$	$.272_{.219-.327}$
Global-Rubric-Tabular	$.600_{.547-.649}$	$.701_{.661-.742}$	$.171_{.135-.208}$	$.266_{.209-.324}$

Table 13: Lupus. AUROC and AUPRC with 95% bootstrap CI. Best per column for each setting is highlighted. Embeddings are from Qwen3-8B; sample sizes refer to per-class training examples.

	AUROC		**AUPRC**	
Qwen3-8B-Zeroshot	$.500_{.499-.500}$		$.009_{.005-.013}$	
GPT5-Mini-Zeroshot	$.550_{.499-.625}$		$.084_{.007-.235}$	
	$n = 10$	$n = \text{All}$	$n = 10$	$n = \text{All}$
Count-GBM	$.754_{.637-.849}$	$.826_{.734-.901}$	$.051_{.012-.144}$	$.109_{.026-.236}$
CLMBR-T	$.614_{.486-.747}$	$.681_{.588-.768}$	$.016_{.008-.028}$	$.020_{.009-.037}$
NaiveText	$.670_{.575-.767}$	$.695_{.556-.822}$	$.025_{.009-.063}$	$.047_{.013-.112}$
Local-Rubric	$.849_{.780-.907}$	$.801_{.676-.906}$	$.046_{.022-.079}$	$.060_{.024-.113}$
Local-Rubric-Basic	$.720_{.602-.831}$	$.701_{.584-.819}$	$.028_{.012-.054}$	$.029_{.012-.056}$
Local-Rubric-NoInterp	$.788_{.713-.865}$	$.775_{.652-.879}$	$.030_{.015-.051}$	$.040_{.017-.069}$
Global-Rubric-Blind	$.731_{.627-.821}$	$.750_{.650-.831}$	$.031_{.012-.071}$	$.076_{.013-.198}$
Global-Rubric	$.747_{.648-.840}$	$.807_{.708-.892}$	$.089_{.016-.220}$	$.058_{.020-.132}$
Global-Rubric-Auto	$.804_{.727-.872}$	$.818_{.739-.886}$	$.033_{.015-.058}$	$.075_{.020-.186}$
Global-Rubric-Tabular	$.727_{.668-.760}$	$.804_{.716-.889}$	$.017_{.009-.025}$	$.043_{.017-.076}$

Table 14: Pancreatic cancer. AUROC and AUPRC with 95% bootstrap CI. Best per column for each setting is highlighted. Embeddings are from Qwen3-8B; sample sizes refer to per-class training examples.

	AUROC		**AUPRC**	
Qwen3-8B-Zeroshot	$.630_{.574-.694}$		$.086_{.043-.155}$	
GPT5-Mini-Zeroshot	$.604_{.554-.662}$		$.205_{.099-.314}$	
	$n = 10$	$n = \text{All}$	$n = 10$	$n = \text{All}$
Count-GBM	$.574_{.503-.644}$	$.933_{.902-.959}$	$.034_{.023-.049}$	$.406_{.258-.546}$
CLMBR-T	$.669_{.589-.740}$	$.812_{.741-.876}$	$.114_{.046-.205}$	$.276_{.165-.407}$
NaiveText	$.697_{.630-.766}$	$.860_{.803-.908}$	$.120_{.051-.201}$	$.321_{.202-.449}$
Local-Rubric	$.843_{.776-.900}$	$.909_{.865-.947}$	$.325_{.206-.454}$	$.491_{.362-.622}$
Local-Rubric-Basic	$.829_{.757-.886}$	$.889_{.836-.932}$	$.268_{.154-.404}$	$.417_{.283-.548}$
Local-Rubric-NoInterp	$.878_{.826-.924}$	$.911_{.866-.946}$	$.306_{.192-.414}$	$.482_{.353-.611}$
Global-Rubric-Blind	$.812_{.755-.864}$	$.864_{.805-.916}$	$.207_{.109-.318}$	$.232_{.134-.346}$
Global-Rubric	$.767_{.691-.839}$	$.874_{.813-.928}$	$.223_{.127-.344}$	$.438_{.301-.575}$
Global-Rubric-Auto	$.718_{.641-.794}$	$.866_{.806-.915}$	$.136_{.071-.217}$	$.335_{.205-.464}$
Global-Rubric-Tabular	$.699_{.633-.763}$	$.847_{.780-.906}$	$.056_{.035-.081}$	$.337_{.217-.473}$

Table 15: Anemia. AUROC and AUPRC with 95% bootstrap CI. Best per column for each setting is highlighted. Embeddings are from Qwen3-8B; sample sizes refer to per-class training examples.

	AUROC		**AUPRC**	
Qwen3-8B-Zeroshot	$.500_{.478-.523}$		$.495_{.468-.520}$	
GPT5-Mini-Zeroshot	$.530_{.510-.549}$		$.514_{.490-.538}$	
	$n = 10$	$n = \text{All}$	$n = 10$	$n = \text{All}$
Count-GBM	$.563_{.538-.587}$	$.562_{.537-.587}$	$.540_{.511-.571}$	$.533_{.502-.563}$
CLMBR-T	$.646_{.624-.669}$	$.821_{.803-.837}$	$.634_{.602-.668}$	$.788_{.759-.816}$
NaiveText	$.581_{.554-.605}$	$.649_{.625-.673}$	$.573_{.541-.607}$	$.634_{.602-.667}$
Local-Rubric	$.584_{.559-.609}$	$.752_{.731-.773}$	$.588_{.554-.622}$	$.710_{.677-.740}$
Local-Rubric-Basic	$.618_{.593-.641}$	$.721_{.698-.743}$	$.606_{.573-.639}$	$.686_{.653-.719}$
Local-Rubric-NoInterp	$.620_{.596-.644}$	$.736_{.714-.758}$	$.619_{.587-.650}$	$.701_{.669-.733}$
Global-Rubric-Blind	$.662_{.638-.686}$	$.776_{.756-.797}$	$.622_{.587-.652}$	$.724_{.691-.757}$
Global-Rubric	$.680_{.657-.702}$	$.764_{.744-.785}$	$.656_{.623-.689}$	$.728_{.696-.758}$
Global-Rubric-Auto	$.638_{.615-.661}$	$.769_{.750-.790}$	$.613_{.582-.644}$	$.734_{.703-.764}$
Global-Rubric-Tabular	$.621_{.598-.644}$	$.787_{.768-.809}$	$.586_{.558-.614}$	$.745_{.712-.776}$

Table 16: Hyperkalemia. AUROC and AUPRC with 95% bootstrap CI. Best per column for each setting is highlighted. Embeddings are from Qwen3-8B; sample sizes refer to per-class training examples.

	AUROC		**AUPRC**	
Qwen3-8B-Zeroshot	$.758_{.738-.779}$		$.735_{.705-.763}$	
GPT5-Mini-Zeroshot	$.786_{.766-.805}$		$.744_{.717-.771}$	
	$n = 10$	$n = \text{All}$	$n = 10$	$n = \text{All}$
Count-GBM	$.568_{.543-.593}$	$.665_{.640-.687}$	$.561_{.530-.595}$	$.644_{.612-.677}$
CLMBR-T	$.562_{.536-.588}$	$.752_{.730-.774}$	$.568_{.535-.601}$	$.763_{.735-.791}$
NaiveText	$.592_{.566-.617}$	$.748_{.726-.769}$	$.581_{.548-.616}$	$.750_{.719-.779}$
Local-Rubric	$.782_{.761-.803}$	$.832_{.814-.850}$	$.773_{.740-.803}$	$.821_{.791-.848}$
Local-Rubric-Basic	$.679_{.654-.702}$	$.812_{.792-.831}$	$.683_{.652-.715}$	$.795_{.764-.824}$
Local-Rubric-NoInterp	$.735_{.712-.758}$	$.830_{.812-.846}$	$.724_{.691-.755}$	$.827_{.803-.852}$
Global-Rubric-Blind	$.660_{.635-.685}$	$.806_{.787-.826}$	$.643_{.608-.676}$	$.798_{.767-.826}$
Global-Rubric	$.681_{.656-.706}$	$.825_{.805-.844}$	$.651_{.618-.684}$	$.818_{.791-.846}$
Global-Rubric-Auto	$.760_{.738-.780}$	$.824_{.806-.842}$	$.764_{.735-.790}$	$.812_{.784-.838}$
Global-Rubric-Tabular	$.635_{.610-.659}$	$.840_{.823-.857}$	$.611_{.581-.642}$	$.828_{.800-.855}$

Table 17: Hypoglycemia. AUROC and AUPRC with 95% bootstrap CI. Best per column for each setting is highlighted. Embeddings are from Qwen3-8B; sample sizes refer to per-class training examples.

	AUROC		**AUPRC**	
Qwen3-8B-Zeroshot	$.687_{.665-.710}$		$.662_{.629-.693}$	
GPT5-Mini-Zeroshot	$.662_{.636-.687}$		$.650_{.617-.683}$	
	$n = 10$	$n = \text{All}$	$n = 10$	$n = \text{All}$
Count-GBM	$.571_{.543-.598}$	$.591_{.563-.621}$	$.561_{.526-.597}$	$.596_{.557-.634}$
CLMBR-T	$.611_{.584-.639}$	$.777_{.755-.799}$	$.595_{.560-.631}$	$.764_{.730-.797}$
NaiveText	$.587_{.559-.616}$	$.680_{.654-.706}$	$.584_{.546-.621}$	$.675_{.638-.709}$
Local-Rubric	$.670_{.644-.698}$	$.780_{.757-.803}$	$.687_{.651-.723}$	$.767_{.732-.798}$
Local-Rubric-Basic	$.642_{.615-.668}$	$.759_{.736-.782}$	$.630_{.593-.667}$	$.745_{.710-.781}$
Local-Rubric-NoInterp	$.709_{.683-.734}$	$.775_{.753-.797}$	$.703_{.668-.737}$	$.767_{.736-.798}$
Global-Rubric-Blind	$.685_{.656-.713}$	$.769_{.746-.795}$	$.707_{.671-.741}$	$.750_{.713-.784}$
Global-Rubric	$.592_{.565-.621}$	$.794_{.774-.816}$	$.596_{.557-.632}$	$.783_{.750-.814}$
Global-Rubric-Auto	$.586_{.558-.614}$	$.727_{.703-.752}$	$.563_{.525-.601}$	$.733_{.697-.763}$
Global-Rubric-Tabular	$.641_{.613-.669}$	$.754_{.729-.776}$	$.603_{.568-.638}$	$.741_{.705-.775}$

Table 18: Hyponatremia. AUROC and AUPRC with 95% bootstrap CI. Best per column for each setting is highlighted. Embeddings are from Qwen3-8B; sample sizes refer to per-class training examples.

	AUROC		**AUPRC**	
Qwen3-8B-Zeroshot	$.686_{.666-.707}$		$.655_{.626-.683}$	
GPT5-Mini-Zeroshot	$.706_{.685-.727}$		$.657_{.629-.687}$	
	$n = 10$	$n = \text{All}$	$n = 10$	$n = \text{All}$
Count-GBM	$.518_{.491-.545}$	$.539_{.514-.563}$	$.529_{.497-.561}$	$.526_{.495-.558}$
CLMBR-T	$.548_{.522-.572}$	$.658_{.633-.681}$	$.557_{.523-.589}$	$.641_{.608-.674}$
NaiveText	$.514_{.489-.538}$	$.595_{.571-.619}$	$.522_{.490-.554}$	$.595_{.563-.626}$
Local-Rubric	$.663_{.638-.688}$	$.740_{.718-.763}$	$.619_{.587-.650}$	$.703_{.668-.737}$
Local-Rubric-Basic	$.561_{.538-.587}$	$.707_{.684-.729}$	$.557_{.526-.591}$	$.686_{.655-.719}$
Local-Rubric-NoInterp	$.549_{.525-.575}$	$.745_{.725-.767}$	$.523_{.493-.553}$	$.725_{.695-.758}$
Global-Rubric-Blind	$.559_{.534-.586}$	$.706_{.684-.729}$	$.568_{.536-.600}$	$.690_{.659-.722}$
Global-Rubric	$.574_{.550-.599}$	$.719_{.697-.740}$	$.564_{.535-.596}$	$.703_{.672-.735}$
Global-Rubric-Auto	$.608_{.583-.633}$	$.733_{.713-.756}$	$.599_{.565-.632}$	$.720_{.691-.751}$
Global-Rubric-Tabular	$.517_{.492-.541}$	$.776_{.757-.797}$	$.533_{.503-.564}$	$.757_{.730-.787}$

Table 19: Thrombocytopenia. AUROC and AUPRC with 95% bootstrap CI. Best per column for each setting is highlighted. Embeddings are from Qwen3-8B; sample sizes refer to per-class training examples.

	AUROC		**AUPRC**	
Qwen3-8B-Zeroshot	$.705_{.683-.725}$		$.637_{.608-.666}$	
GPT5-Mini-Zeroshot	$.739_{.719-.759}$		$.670_{.645-.696}$	
	$n = 10$	$n = \text{All}$	$n = 10$	$n = \text{All}$
Count-GBM	$.498_{.473-.525}$	$.556_{.531-.580}$	$.507_{.477-.540}$	$.551_{.519-.582}$
CLMBR-T	$.522_{.496-.548}$	$.627_{.604-.654}$	$.523_{.492-.554}$	$.606_{.575-.640}$
NaiveText	$.518_{.491-.545}$	$.612_{.586-.635}$	$.524_{.491-.555}$	$.588_{.556-.622}$
Local-Rubric	$.771_{.749-.791}$	$.841_{.824-.859}$	$.732_{.701-.765}$	$.810_{.781-.836}$
Local-Rubric-Basic	$.608_{.584-.632}$	$.760_{.739-.781}$	$.572_{.541-.606}$	$.742_{.711-.770}$
Local-Rubric-NoInterp	$.718_{.696-.739}$	$.815_{.796-.834}$	$.673_{.638-.705}$	$.769_{.736-.800}$
Global-Rubric-Blind	$.611_{.588-.634}$	$.828_{.810-.847}$	$.583_{.552-.617}$	$.804_{.776-.830}$
Global-Rubric	$.625_{.601-.648}$	$.851_{.834-.868}$	$.585_{.554-.617}$	$.834_{.807-.859}$
Global-Rubric-Auto	$.616_{.592-.640}$	$.828_{.809-.846}$	$.593_{.563-.625}$	$.800_{.770-.829}$
Global-Rubric-Tabular	$.656_{.632-.679}$	$.860_{.844-.875}$	$.613_{.584-.640}$	$.841_{.818-.866}$

Table 20: Chest X-ray. AUROC and AUPRC with 95% bootstrap CI. Best per column for each setting is highlighted. Embeddings are from Qwen3-8B; sample sizes refer to per-class training examples.

	AUROC		**AUPRC**	
Qwen3-8B-Zeroshot	$.546_{.519-.573}$		$.526_{.491-.560}$	
GPT5-Mini-Zeroshot	$.520_{.498-.541}$		$.507_{.476-.538}$	
	$n = 10$	$n = \text{All}$	$n = 10$	$n = \text{All}$
Count-GBM	$.571_{.541-.602}$	$.609_{.576-.642}$	$.543_{.502-.583}$	$.582_{.540-.624}$
CLMBR-T	$.453_{.421-.484}$	$.609_{.578-.640}$	$.465_{.427-.500}$	$.600_{.559-.643}$
NaiveText	$.558_{.533-.584}$	$.616_{.592-.640}$	$.551_{.519-.583}$	$.609_{.577-.641}$
Local-Rubric	$.517_{.492-.543}$	$.606_{.583-.630}$	$.503_{.474-.535}$	$.605_{.571-.638}$
Local-Rubric-Basic	$.487_{.457-.519}$	$.599_{.568-.628}$	$.493_{.454-.531}$	$.594_{.554-.633}$
Local-Rubric-NoInterp	$.525_{.491-.556}$	$.596_{.565-.625}$	$.509_{.469-.546}$	$.581_{.537-.625}$
Global-Rubric-Blind	$.437_{.413-.461}$	$.585_{.559-.608}$	$.455_{.427-.486}$	$.587_{.556-.620}$
Global-Rubric	$.513_{.489-.538}$	$.594_{.567-.619}$	$.521_{.489-.552}$	$.582_{.550-.614}$
Global-Rubric-Auto	$.503_{.470-.534}$	$.575_{.543-.605}$	$.497_{.460-.534}$	$.557_{.518-.596}$
Global-Rubric-Tabular	$.500_{.500-.500}$	$.540_{.507-.572}$	$.497_{.470-.523}$	$.527_{.486-.567}$

D Full Prompts Used in Rubric Representation Learning Methods

D.1 Prompt to LLM for Global Rubric Creation (Figure 3, Panel B)

```
# Prompt used with GPT-5-mini for global rubric synthesis
---
You are a medical expert designing a structured rubric for a clinical
prediction task.

## Task
- Name: {task_name}
- Query: {task_query}

## Context
You will be given {40} labeled patient EHR examples ({20} positive, {20}
negative). Another model will later use your rubric to transform new
patient EHRs into structured summaries, which will then serve as input to
a supervised classifier.

## What You Must Do
Study the examples below. Combine what you observe in them with your medical
knowledge to design a rubric template - a set of named fields that, when
filled in for any patient, produce a structured summary optimized for this
prediction task.

The rubric should:
1. **Be data-driven and discriminative.** Identify which features, patterns,
and interactions actually separate the positive and negative cases. The
rubric should capture not just obvious indicators but also subtler or
compound features you notice. At the same time, do not overfit to these
40 cases - use your clinical knowledge to include factors that are generally
relevant even if not prominent in this sample.

2. **Be structured and consistent.** Every rubricified output must follow
the same field names and order. For each field, specify what to extract from
the EHR and how to format it. Specify what to write when data is absent.

3. **Extract facts only.** The evaluator filling in the rubric must extract
and organize information from the EHR. It must NOT make predictions, assign
risk levels, or draw conclusions.

4. **Be concise.** The rubric should focus on extracting information that
is relevant to the task. It should not ask the evaluator to reproduce the
entire EHR.

## Positive Examples (Ground Truth: Yes)
{NaiveText EHR serializations of 20 positive examples concatenated (x_text
format)}

## Negative Examples (Ground Truth: No)
{NaiveText EHR serializations of 20 negative examples concatenated (x_text
format)}

## Output
Output ONLY the rubric template itself - the instructions another model
will follow to transform a patient EHR. No preamble, no explanation of your
reasoning. The template must be self-contained and directly usable.
```

Figure 14: Prompt used with GPT-5-mini to guide global rubric creation from NaiveText serializations (x_{text}) of EHR examples.

D.2 Prompt to LLM for Global Rubric Application (Figure 3, Panel D)

```
# Prompt used with GPT-5-mini for global rubric application
```

```
You are a medical data extraction specialist.

## Task
{task_query}

## Rubric Template (follow this exactly)
{rubric_instructions}

## Patient EHR
{ehr_text ($x_{\text{text}}$ format)}

## Instructions
Fill in every field of the rubric template above using ONLY information from
this patient's EHR.

Rules:
- Follow the exact field order and section structure of the rubric.
- Be concise:  use short phrases, numbers, and dates.  Do not write
paragraphs.
- If data for a field is not present in the EHR, write "No data".
- Do NOT add commentary, predictions, risk assessments, or conclusions.
- Do NOT include any information not found in the EHR above.

Rubric output:
```

Figure 15: Prompt used with GPT-5-mini for transforming a naive text serialized input (x^{text}) into its rubric text serialization version (x^{rubric}).

D.3 Prompt to LLM for Creating a Global Rubric Application Parser (Figure 3, Panel E)

```
# Prompt used with GPT-5.2 for generating a parser script for rubric application
---
You are an expert Python developer and medical informaticist.

## Your Task
Write a complete, self-contained Python script that reads patient EHR serializations and fills in
a structured clinical rubric template using **deterministic string/regex parsing only** -- no LLM
API calls, no network requests.

## Clinical Task Context
- Task name: {task_name}
- Prediction query: {task_query}

## Rubric Template to Fill
The script must fill in every field defined in the following rubric instructions:
{rubric_instructions, ℛ}

## EHR Serialization Format
Below are 40 example patient EHR serializations from the training cohort, labeled by ground-truth
outcome. For each patient you are shown BOTH:
1. The raw naive text EHR serialization.
2. The LLM-produced rubric fill for that exact patient -- showing you how the fields should be
extracted from the raw text.

Use these paired examples to understand the extraction mapping precisely.
{40 paired examples of naive text serializations (x^text) and LLM-filled rubric text serializations
(x^rubric).}

## Required Script Interface
The generated script must:
1. Accept the following command-line arguments via argparse:
- `-input_dir` : root directory of naivetext serializations - `-output_dir` : root directory for
llmrubric-parser outputs - `-task` : task name - `-splits` : one or more of `train val test`

2. For each split, read `{{input_dir}}/{{task}}/{{split}}.json` -- a JSON array where each
element has:
- `patient_id` (int) - `prediction_time` (ISO datetime string) - `task` (str) - `split` (str) -
`label` (bool) - `serialization` (str) ← the EHR text to parse

3. For each patient call `fill_rubric(serialization: str) -> str`, which:
- Extracts all rubric fields from the EHR text using regex and string operations - Returns a
filled-in rubric string that follows the exact field names, order, and format from the rubric
template above - Writes "NA" for any field whose data is absent from the EHR

4. Write output to `{{output_dir}}/{{task}}/{{split}}.json` -- a JSON array where each element
has:
- `patient_id` (int) - `prediction_time` (str) - `task` (str) - `split` (str) - `label` (bool) -
`rubricified_text` (str) ← output of fill_rubric()

5. Create output directories as needed (parents=True, exist_ok=True).

6. Print progress to stdout: total patients processed per split.

## Constraints
- Use only Python standard library plus `re`, `json`, `argparse`, `pathlib`, `sys`. No third-party
packages. - No LLM API calls, network requests, external tools. - The `fill_rubric` function
must be deterministic and handle missing data gracefully (write "NA" rather than raising
exceptions). - The script must be syntactically valid Python 3.8+. - Do NOT hardcode file
paths -- use the argparse arguments.

## Output
Output ONLY the Python script, with no explanation, no preamble, and no markdown fences. The
output must start with `#!/usr/bin/env python3` and be directly writable to a .py file.
```

Figure 16: Prompt used with GPT-5.2 to create a parser script for transforming a naive text serialized input (x^{text}) into its rubric text serialization version (x^{rubric}).

D.4 Prompt to LLM for Creating a Global Rubric Tabularization Parser (Figure 3, Panel F)

```
# Prompt used with GPT-5.2 for generating a parser script to transform rubric serializations to
tabular features
---
You are an expert Python developer and medical informaticist.

## Your Task
Write a complete, self-contained Python featurizer script that reads rubric-formatted patient
EHR texts and converts each one into a **fixed-dimension numeric feature vector** using
deterministic string/regex parsing - no LLM calls, no network requests.

## Clinical Task Context
- Task name: {task_name}
- Prediction query: {task_query}

## Rubric Parser Source (shows all rubric field names and their text formats)
The following is the parser that generates the rubric text. Study it to understand which
fields exist and how their values are formatted in the text. This is the **ground truth** for
what fields can appear in a rubric text and how their values are formatted.

```python
{task-specific rubric parser generated via prompt in Section D.3}
```

Reference Rubric Texts ({20} positive, {20} negative)

Important context: These {40} patients are the cohort that was used to *design* the rubric
itself. They are provided as examples so you can calibrate your regex patterns against actual
data.

However, the featurizer you write will be applied to a **much larger dataset** (thousands
of patients). Your feature extraction logic must therefore be:

- **General**: handle any value the rubric parser could plausibly produce, not just the values
seen in these 40 patients

- **Robust**: gracefully handle missing, NA, or unexpected values for every field

- **Comprehensive**: derive features from every field in the rubric, even if that field
happens to be NA for all {40} examples shown here

Use the parser source above as the authoritative specification of fields and value formats; use
the examples below to validate and calibrate your regex patterns.

{40 example text serialization in x^rubric format}

Required Script Interface

The generated script must:
1. Accept CLI arguments via argparse: - `-input_dir`
- `-output_dir`
- `-task`
- `-splits`

2. For each split, read `{{input_dir}}/{{split}}/{{task}}.json` - a JSON array where each
element has:
- `patient_id` (int)
- `label_time` (ISO datetime string)
- `label_value` (bool)
- `conversations` (list) - rubric text is in `conversations[1]["content"]` between `-- Patient
EHR --` and `-- End of EHR --`

3. Implement `def extract_features(rubric_text: str) -> dict[str, float]`:
- Parse every rubric field from the text
- Return a flat dict mapping feature name → float value
- For **numeric fields**: extract the number; if missing/NA write `0.0` and set
`{{field}}_missing = 1.0`
- For **categorical / Yes/No fields**: one-hot encode all known values; unknown/NA → all zeros
plus a `{{field}}_missing = 1.0` indicator
- All returned values must be float (0.0 or 1.0 for binary, numeric otherwise)
- The dict must have the **same keys in the same order** for every call (fixed schema)

4. Define `SCHEMA: list[dict]` at module level - one entry per feature with keys:
- `"name"`: feature name (matches key in extract_features output)
- `"type"`: `"numeric"`, `"binary"`, or `"categorical"`
```

```
- `"description"`:  short human-readable description
- `"possible_values"`:  list of string values for categorical/binary fields, omit for numeric

5.  For each split, build an N×F float32 matrix from `extract_features`, save as:
- `{{output_dir}}/{{task}}/{{split}}.npz` with numpy keys:
- `embeddings`:  shape (N, F) float32
- `labels`:  shape (N,) int32
- `patient_ids`:  shape (N,) int64
- `prediction_times`:  shape (N,) object (strings)

6.  Save `{{output_dir}}/{{task}}/feature_schema.json` once (after processing the first split):
```json
{{ "task":  "{task}",
"task_query":  "{task_query}",
"num_features":  <F>,
"features":  <SCHEMA list>
}}
```

7. Create output directories as needed. Print progress to stdout.

Constraints
- Use only Python standard library plus `re`, `json`, `numpy`, `argparse`, `pathlib`, `sys`.
No third-party packages beyond numpy.
- No LLM API calls, no network requests.
- `extract_features` must be deterministic and never raise exceptions on any input (catch all
errors, default to 0.0).
- The script must be syntactically valid Python 3.8+.
- Do NOT hardcode file paths - use the argparse arguments.
- Aim for **at least 30 features** to capture the richness of the rubric.
Include all numeric fields, all categorical fields (one-hot), and Yes/No procedure/comorbidity
flags.

Output
Output ONLY the Python script, with no explanation, no preamble, and no markdown fences. Start
with `#!/usr/bin/env python3`.
```

Figure 17: Prompt used with GPT-5.2 to create a parser script that transforms rubric-transformed inputs (x^{rubric}) into a fixed-dimensional tabular feature vector.

D.5 Prompt to LLM for Stripping Away Interpretation of Evidence from a Local Rubric Representation (Figure 4, Left)

```
# Prompt used with GPT-5-mini for stripping away interpretive language from
Local-Rubrics
---
Below is a clinical summary of a patient's EHR. Your task is to produce
a MINIMALLY EDITED version that removes ONLY interpretive and predictive
language.

-- START OF EHR DATA --
{NaiveText_Serialization (x^text)}
-- END OF EHR DATA --

REMOVE (by deleting or replacing with neutral phrasing):

- Causal/risk language ("increases risk", "raises concern", "associated with
worse outcomes")
- Protective framing ("protective", "favorable", "reassuring", "reduces risk")
- Interpretive connectives ("suggests", "indicates", "consistent with",
"likely", "concerning for")
- Weighing/aggregating/conclusion statements that assess overall risk or
probability
- Predictive statements or probability assessments

KEEP UNCHANGED as much as possible:

- The overall structure, section ordering, and formatting (including think
tags and ### headers)
- All factual clinical content:  demographics, diagnoses, lab values with
numbers, vitals, medications, procedures, dates, and other concrete data
points
- The exact wording of factual statements - do NOT rephrase facts that contain
no interpretation
- Section headers - keep them but strip interpretive framing from bullet
points beneath them
- Bullet point structure and ordering

IMPORTANT:

- Make the FEWEST changes necessary.  If a bullet point is purely factual,
leave it verbatim.
- If a bullet mixes fact + interpretation, keep the fact and delete only
the interpretive clause.  Example:  "Hyperglycemia (glucose 202 mg/dL)
indicating diabetes or stress hyperglycemia (associated with worse outcomes)"
-> "Hyperglycemia (glucose 202 mg/dL)."
- If a bullet is purely interpretive with no factual content, delete it
entirely.
- Do NOT add any new information, reorganize sections, or change the summary's
structure.
- The section "WEIGHING AND AGGREGATING THE EVIDENCE" (section 5) should
be ENTIRELY removed - delete the header and all its content.  It is purely
interpretive.
```

Figure 18: Prompt used with GPT-5-mini for stripping away interpretive language from Local-Rubric representation (*i.e.,* modifying x^{rubric}).

E Reproduction and compute costs

We summarize the practical resource requirements for reproducing our experiments on the subsampled EHRSHOT splits described above (Section B, Table 1).

NaiveText truncation. Following Hegselmann et al. [15], we clip NaiveText serializations at 8,192 tokens (Qwen3-8B tokenizer).

LLM API costs. Methods with a per-example LLM call—Global-Rubric, Global-Rubric-Blind, Local-Rubric, and Local-Rubric-Basic—each cost approximately $50 per task in GPT5-Mini API usage on our subsampled splits, totaling roughly $3,000 across 4 methods and 15 tasks. Global-rubric synthesis and the one-time parser/tabularization scripts used by Global-Rubric-Auto and Global-Rubric-Tabular (Figure 3, Panels E–F) add a small one-off cost; once those scripts are produced, applying them to new examples at deployment time is essentially free.

Embedding compute. Text embeddings (`Qwen3-Embedding-8B` for all main results, plus the open-weights backbones used in Section C.5) are computed on a node with 4 NVIDIA A100 GPUs. Embedding the full dataset for any given representation method (NaiveText or any rubric variant) takes a few hours per method.

Downstream training. The downstream classifiers (logistic regression on frozen embeddings; LightGBM for Count-GBM; XGBoost for Global-Rubric-Tabular) are lightweight and run on commodity CPU hardware in seconds to minutes per task; their cost is negligible compared to the embedding and LLM-call steps above.

F Case Study: Learned Tabular Features for Prediction of Hyponatremia

We provide qualitative analysis of the learned tabular features (Figure 3, Panel F) for prediction of hyponatremia abnormality. We find that across the 15 tasks, the auto-generated rubric feature schemas range from 147 to 450 features per task. Most tasks cluster around 200–250 features. The features are predominantly binary (72%), followed by numeric (19%) and categorical (9%). The high binary share reflects pervasive one-hot encoding of categoricals and the inclusion of a `_missing` indicator for nearly every field. Numeric features capture lab values, vitals, and counts.

We focus on learned tabular features for the hyponatremia lab task, for which the full global rubric is given in Appendix G.2. We make several observations. The feature structure closely mirrors the diagnostic decision tree used clinically when evaluating hyponatremia. A first step in clinical reasoning is determining whether apparent hyponatremia is physiologic or artificially low due to hyperglycemia or other osmotic effects; accordingly, the rubric extracts recent glucose measurements (`Glucose-Last3`) and serum osmolality values, while the tabular features include indicators reflecting level of glucose in the blood, *e.g.* `glucose-type-blood-present`. If true hypotonic hyponatremia is present, clinicians next evaluate urine osmolality and urine sodium to distinguish between states of antidiuretic hormone (ADH) activity and renal sodium handling, which helps identify etiologies such as SIADH or hypovolemia [55]. Consistent with this framework, the rubric extracts urine sodium, urine osmolality, and conditions associated with SIADH (*e.g.*, pulmonary infections, CNS disorders, malignancy). The resulting features include indicators that reflect such conditions, *e.g.*, `acute-cond-Pulmonary infection / pneumonia / pulmonary disease` and `acute-cond-any`. Thus, the learned features directly operationalize the same diagnostic flow in clinical practice.

A second group of features captures baseline risk factors and comorbidities that predispose patients to hyponatremia. The rubric explicitly extracts conditions such as chronic kidney disease (CKD), dialysis history, malignancy, and medications known to induce hyponatremia (*e.g.,* thiazide diuretics). These signals appear directly in the tabularized representation through features such as `dialysis-history-Yes`, `procedure-Hemodialysis`, and `med-class-count-thiazide-diuretic`. These variables correspond to well-known clinical risk factors for hyponatremia, including impaired renal free-water handling and medication-induced sodium loss [62].

Finally, the most predictive signals arise from the acuity and trajectory of prior sodium measurements. The rubric explicitly extracts the three most recent sodium values and the lowest sodium in the prior 90 days, along with contextual metadata such as the setting of the measurement (*e.g.,* inpatient vs outpatient). Correspondingly, the most important coefficients in the tabular feature set correspond to prior sodium measurements, including `serum-na-recent-le-134` and `prior-documented-hyponatremia-Yes`. Clinically, this is expected, as patients with a history of chronic or recurrent hyponatremia (*e.g.*, due to heart failure or cirrhosis) are substantially more likely to have abnormal sodium levels on subsequent laboratory testing [60].

This example illustrates how rubric representations surface task-relevant information that would otherwise be buried in a long, heterogeneous text-serialization of the patient record. In the naive text format, prior sodium measurements appear scattered across multiple visits and lab panels, interleaved with unrelated clinical events. The rubric reorganizes this information into a compact set of fields that explicitly capture the recent trajectory of the lab. In doing so, it converts diffuse signals in language space into structured features that a simple downstream model can use efficiently.

G Full Global Rubric Examples

G.1 Full Global Rubric for the Hypertension Diagnosis Task

```
RUBRIC INSTRUCTIONS FOR TASK: HYPERTENSION
---
Rubric purpose
- Provide a reproducible, stepwise process to transform any EHR into a structured,
clinical-evidence summary useful for assessing the likelihood that a patient will develop
hypertension in the next year.
- The rubric standardizes what to extract, how to summarize trends and risk factors, and how
to record uncertainty and provenance so downstream models or clinicians can apply consistent
reasoning.


How to use this rubric
- Follow the numbered extraction and analysis steps for each new patient.
- Populate the structured template fields exactly (use units shown). If data are missing,
enter "missing" and note time windows attempted.
- Do NOT make a final yes/no prediction inside the form. Instead, produce the structured
summary and quantitative or qualitative risk-domain scores for downstream modeling.


A. Preparation (before extracting)
1. Define the prediction window: "next year" relative to the EHR reference date and time.
2. Define time windows to extract:
- Very recent: last 30 days
- Recent: 31-180 days
- Baseline/remote: >180 days up to available history
3. Standardize units and formats:
- Blood pressure: mmHg (systolic/diastolic)
- Weight: kg or oz → convert to kg if numeric calculations needed
- Height: cm or in → convert to meters
- Labs: use usual clinical units (creatinine mg/dL, A1c %, etc.)
4. Log data sources (vitals, problem list, medications, laboratory, procedures, notes) and
timestamp of extraction.


B. Step-by-step extraction & transformation procedure

Step 1 -- Demographics and baseline context
- Extract:
- Age (years)
- Sex / gender
- Race / ethnicity (if available)
- Relevant social history: tobacco (current/former/never), alcohol (heavy/regular/rare/none),
illicit drug use, tobacco product types
- Pregnancy status (current or past complications such as pre-eclampsia)
- Baseline height and weight; calculate BMI (kg/m^2) and BMI category
- Record date of last update for each demographic item.

Step 2 -- Blood pressure (BP) data extraction and normalization
- Extract all systolic and diastolic BP values with timestamps and context (office, inpatient,
ED, home, ambulatory, perioperative).
- Normalize values: remove implausible readings (document them) and ensure mmHg units.
- For each time window:
- Compute count, mean, median, standard deviation, minimum, and maximum.
- Identify last available BP and date.
- Flag highest recent systolic and diastolic values with dates.
- Compute trend metrics:
- Recent slope = (mean_recent − mean_baseline) / time (mmHg per month); indicate direction only
if clinically meaningful (e.g., ≥3 mmHg/year).
- BP variability indicator: SD of systolic BP in recent window; flag high variability if SD
>10 mmHg.
- Categorize BP using ACC/AHA thresholds:
- Normal (<120/<80)
- Elevated (120-129/<80)
- Stage 1 Hypertension (130-139 or 80-89)
- Stage 2 Hypertension (≥140 or ≥90)
- If mixed, note "discordant" and list counts per category.

Step 3 -- Antihypertensive and BP-impacting medications
- Extract current and recent medications with start and stop dates if available.
- Flag antihypertensives (ACEi, ARBs, beta-blockers, diuretics, CCBs, vasodilators).
- Flag BP-raising agents (systemic corticosteroids, NSAIDs, decongestants, stimulants,
calcineurin inhibitors, SNRIs, MAOIs, some oral contraceptives).
- For each flagged medication record name, dose, dates, indication, and temporal relation to BP
```

```
changes.

Step 4 -- Comorbidities associated with increased HTN risk
- Extract diagnoses and ICD codes with dates:
- Major risk:  CKD, diabetes, CVD, PAD, OSA, endocrine causes, pregnancy or pre-eclampsia,
obesity (BMI ≥30), heavy alcohol use.
- Moderate risk:  hyperlipidemia, metabolic syndrome, thyroid disease, autoimmune disease with
renal involvement.
- Secondary HTN clues:  resistant BP, hypokalemia, episodic symptoms.
- Record first documentation date, last active date, and severity when available.

Step 5 -- Relevant laboratory data
- Extract labs with dates grouped by time window:
- Creatinine / eGFR
- Electrolytes (Na, K, HCO3)
- Glucose, HbA1c
- Lipids
- Urine albumin or protein
- Thyroid tests
- Aldosterone/renin, cortisol, catecholamines if available
- Flag abnormal values with interpretation (e.g., eGFR <60 ml/min).

Step 6 -- Procedures and objective testing
- Extract echocardiography, renal imaging, sleep studies, ABPM.
- Note evidence of end-organ effects (LVH, albuminuria, renal disease).

Step 7 -- Social, behavioral, and family data
- Smoking status and intensity.
- Alcohol use severity.
- Family history of HTN or early CVD.
- Adherence or socioeconomic barriers if documented.

Step 8 -- Acute confounders
- Identify acute illness, pain, surgery, sepsis, AKI, or inpatient context affecting BP
interpretation.
- Avoid using isolated inpatient readings without outpatient corroboration.

Step 9 -- Domain synthesis and scoring
- For each domain, record evidence, recency, and confidence (High/Moderate/Low).
- Domains include BP phenotype, medications, metabolic risk, kidney function, secondary HTN,
end-organ disease, behavior, and acute confounders.
- Assign severity (Major/Moderate/Minor) and create a domain scorecard; do NOT generate a final
binary label.

Step 10 -- Evidence provenance and missing data
- Record source, timestamp, and confidence for each major item.
- Explicitly flag critical missing data (e.g., no outpatient BP in 12 months).

Step 11 -- Structured output
- Produce a standardized summary with demographics, BP summary, medications, comorbidities,
labs, procedures, lifestyle, acute confounders, domain scorecard, missing data, and a 2-4
sentence neutral text summary.

Step 12 -- Guidance notes
- Recommend confirmatory testing or review where appropriate (e.g., home BP, med review,
nephrology referral).
- Do not conclude final risk.


Final note to the user
- Use this rubric to populate the structured template for every patient.  Do not record a
final hypertension risk classification here; the output is intended for downstream models or
clinician judgment.
```

Figure 19: Global rubric instructions for extracting structured patient profiles for 1-year hypertension diagnosis prediction.

G.2 Full Global Rubric for the Hyponatremia Lab Result Prediction Task

```
RUBRIC INSTRUCTIONS FOR TASK: HYPONATREMIA LAB RESULT
---------------------------------------------------------------------------
PredictionDate: [Extract the 'current time' / prediction timestamp from the EHR header].
Format: YYYY-MM-DD. If not present write NA.
Patient:

- Age: [years as integer from EHR]. If not present write NA.
- Sex: [as documented: MALE / FEMALE / Other / Unknown]. If not present write NA.
- Race/Ethnicity: [as documented]. If not present write NA.

ProblemListFlags (presence and dates):

- Chronic kidney disease / End-stage renal disease (CKD/ESRD): [Yes / No]. If Yes, list
documented term(s) and most recent date(s) (YYYY-MM-DD). If none write No.
- Dialysis history/procedure in record: [Yes / No]. If Yes, list procedure name(s) and most
recent date(s). If none write No.
- Prior documented hyponatremia / 'hypo-osmolality and or hyponatremia'': [Yes / No]. If Yes,
give the documentation text and date(s). If none write No.
- Active malignancy listed in Problem List or current visits: [Yes / No]. If Yes, list
malignancy type(s) and most recent date(s). If none write No.

SerumSodium_Last3 (most recent first): For up to 3 most recent serum/plasma/blood sodium
measurements, extract a line per measurement in this exact format:

- YYYY-MM-DD (days_before_prediction): [value] mmol/L ; Specimen=[serum/plasma/blood] ;
Setting=[ED/Inpatient/Outpatient/Lab] ; Note=[any explicit result comment if present]
If fewer than 3 measurements exist, include those available; if none write NA.
SerumSodium_Min90:
- Lowest documented serum/plasma/blood sodium value in the prior 90 days (value mmol/L) and
date (YYYY-MM-DD). If none write NA.
SerumOsmolality_Last3:
- For up to 3 most recent serum osmolality measurements: YYYY-MM-DD (days_before_prediction):
[value] mOsm/kg ; Setting=[as above]
If none write NA.

UrineStudies_Last3:
- For up to 3 most recent urine study sets, extract for each available element on one line:
- YYYY-MM-DD (days_before_prediction): UrineNa=[value] mmol/L ; UrineOsm=[value] mOsm/kg ;
SpecificGravity=[value] ; Setting=[ED/Inpatient/Outpatient/Lab]
Only include elements that are present for that date. If no urine studies documented write NA.

RenalFunction:
- Most recent serum creatinine (mg/dL) and date: YYYY-MM-DD: [value] mg/dL. If none write NA.
- Most recent BUN (mg/dL) and date: YYYY-MM-DD: [value] mg/dL. If none write NA.
- Recent acute renal failure / acute kidney injury entries within 30 days: [Yes / No]. If Yes
include diagnosis text and date(s). If none write No.

VolumeRelatedFindings (documented in problem lists or visit notes within past 30 days):
- Extract presence with dates for these items (list each if present as "Item: YYYY-MM-DD;"):
Edema, Ascites, Hypotension (documented low BP or explicit "hypotension"), Dehydration,
Vomiting, Diarrhea, Nasogastric/feeding tube, Ileostomy/colostomy, Recent large-volume
paracentesis. If none of these documented in past 30 days write NA.

Medications_PotentiallyAffectingSodium (recent administrations - extract from medication list /
inpatient meds / discharge meds):
- Time window: last 14 days before PredictionDate (if EHR supports more granular times use
those). For each relevant med/class present include one line:
- [YYYY-MM-DD last administration if available] : [Medication name] ; Class=[thiazide/loop
diuretic / SSRI / SNRI / TCA / anticonvulsant (carbamazepine/oxcarbazepine) / NSAID / SSRI,
etc.] ; Route=[oral/IV] ; Dose if documented=[text]
- If none of these medication classes documented in last 14 days write NA.
- Also include "Chronic diuretic use noted (Yes/No) and last documentation date" (e.g.,
long-term thiazide).

IVFluids_Last72h:
- List IV fluid administrations in last 72 hours (date/time if available) in the format:
- YYYY-MM-DD: [fluid type as documented, e.g., D5W / D5NS / 0.9% NaCl / hypotonic saline / LR /
"glucose 50 mg/mL prefills"] ; Volume if documented.
- If none documented write NA.
AcuteConditions_AssociatedWithSIADHorHyponatremia (documented within 30 days):
- For each present within 30 days, list as "Condition: YYYY-MM-DD" from problem/visit notes:
- Pulmonary infection / pneumonia / pulmonary disease
- CNS disorder (stroke, hemorrhage, encephalopathy)
- Sepsis / severe infection
```

```
- Recent major surgery / Postoperative state
- Pain / Severe nausea (if explicitly documented)
- Malignancy active (if not already in ProblemListFlags)
If none documented write NA.

RecentProcedures_Chemotherapy_Transfusion (last 30 days):
- List any of:  major surgery, chemotherapy, recent blood transfusion, paracentesis, TPN (total
parenteral nutrition), plasmapheresis, hemodialysis - format:
- YYYY-MM-DD: [procedure name / chemo agent e.g., paclitaxel] ; Notes=[if available]
If none write NA.
Glucose_Last3:
- Up to 3 most recent serum/plasma or point-of-care glucose values (most recent first):
- YYYY-MM-DD: [value] mg/dL ; Type=[serum/plasma/glucometer] ;
Setting=[ED/Inpatient/Outpatient]
If none write NA.

SerumProteinOrLipidExtremes:
- If very high triglycerides or abnormal total protein/albumin documented close to sodium
measurement, extract:
- YYYY-MM-DD: Triglycerides=[value] mg/dL ; TotalProtein=[value] g/dL ; Albumin=[value] g/dL
If none documented write NA.

PriorHyponatremiaHistory:
- Any historical low sodium episodes before 90 days (brief):  list lowest prior value and
date(s) or write NA.

LabQualityNotes:
- Any documented lab-quality flags on sodium measurement (e.g., hemolysis, lipemia, ‘‘evacuated
blood collection tube’’ note, specimen issues):  extract verbatim note and date(s).  If none
write NA.

RelevantVitalSigns_NearMostRecentSodium:
- From the same encounter as the most recent sodium (if identifiable), extract:
systolic/diastolic BP (mmHg), heart rate (bpm), and whether on oxygen or dialysis that
encounter.  Format:
- Date:  YYYY-MM-DD ; SBP=[value] ; DBP=[value] ; HR=[value] ; Oxygen=[yes/no] with O2 sat if
given ; DialysisThisEncounter=[yes/no]
If not available write NA.

FreeText_Findings_Cues:
- Extract any verbatim phrases (short quotes) that explicitly mention hyponatremia-related
language in notes or problem list (e.g., "hyponatremia", "hypo-osmolality", "SIADH", "hypotonic
fluids", "low sodium") with the date and the note type.  Format:
- YYYY-MM-DD ; Source=[ProblemList/VisitNote/LabComment] ; Text="[exact phrase]"
If none write NA.

DataCompleteness:
- For each of the following categories indicate [Present / Absent / Not documented]:  Serum
sodium labs, urine sodium/osmolality, serum osmolality, recent meds list, dialysis record, IV
fluids record, creatinine/BUN. Example:  SerumSodium:  Present ; UrineSodium:  Absent ; etc.

ExtractionRules / Formatting Rules (must follow exactly):
- Always extract facts only; do not add interpretation, risk assessment, or predictions.
- Dates:  use YYYY-MM-DD as in EHR; if EHR provides relative days include "(N days before
prediction)" after date.
- When multiple values on same date, include all values separated by ";".
- If an item not found anywhere in the EHR, write exactly "NA".
- Keep each field on a single line (except the repeated-measure lists which may have up to
three lines as specified).
- Use units exactly as specified (mmol/L for Na and UrineNa; mOsm/kg for osmolality; mg/dL for
glucose/BUN/creatinine; mg/dL for triglycerides).
- Do not synthesize or infer ranges; extract only documented numeric values and verbatim text.

EndOfTemplate.
```

Figure 20: Global rubric instructions for extracting structured patient profiles for hyponatremia lab result prediction (abnormal vs. normal).

H LoRA fine-tuning and chain-of-thought variants

All methods in the main paper keep the LLMs frozen: an LLM produces a textual representation, a frozen text-embedding model encodes it, and only a logistic regression head is trained on the labels. Alternatively, the LLM itself can be used as the classifier, by prompting it for a verdict or by fine-tuning it on the prediction task with low-rank adaptation (LoRA) [17], with or without a chain-of-thought (CoT) reasoning trace preceding the answer. We explored these alternatives during the project, and we summarize our findings here to provide some insight into the LoRA and CoT approaches. We ultimately scaled the embedding approach, as it is much cheaper: each representation is embedded once and reused across training-set sizes, and only a logistic regression head is fit, whereas fine-tuning requires a separate LoRA run for every task and training set.

H.1 Results from an earlier four-task version of the project

Scope of the comparison. The results in this subsection come from an earlier version of the project, where some of our system prompts were still in beta testing, and hence they do not constitute an apples-to-apples comparison with the results in the main paper. The earlier version evaluated four of the six new-diagnosis tasks (acute MI, hyperlipidemia, hypertension, and pancreatic cancer), did not include Local-Rubric-NoInterp, Local-Rubric-Basic, Global-Rubric-Blind, Global-Rubric-Auto, or Global-Rubric-Tabular, and used $n = 16$ training examples (8 per class) in the small-sample regime instead of $n = 10$ per class. Both versions report AUROC and AUPRC on the test split and use the original EHRSHOT training and test splits for these four tasks, so the test sets agree up to a few examples.

Methods. All three variants fine-tune Qwen3-8B [43] with LoRA (rank 16, scaling factor 32) using Unsloth [59], with early stopping based on the loss over a label-balanced set of 100 validation examples. Let $T(\cdot, q)$ denote a task-conditioned prompt for the task query q, similar to the templates in Figure 4, which asks for a one-word answer (`Positive` or `Negative`).

- NaiveText-LoRA fine-tunes the model to predict the label word y from $T(x^{\text{text}}, q)$ via next-token prediction, by minimizing $-\log p_\theta\big(y \mid T(x^{\text{text}}, q)\big)$.
- Global-Rubric-LoRA is identical to NaiveText-LoRA, but replaces x^{text} with the global rubric representation x^{rubric}.
- SFT-CoT fine-tunes the model to generate a reasoning trace z followed by the label, by minimizing $-\log p_\theta\big(z, y \mid T(x^{\text{text}}, q)\big)$, similar to the supervised reasoning training of Lin et al. [31]. We synthesize the training traces by prompting GPT5-Mini with x^{text} and the ground-truth outcome, instructing it to reason from the EHR as if the outcome were unknown. The traces follow a fixed section structure close to that of the Local-Rubric prompt (Figure 4, left), with a final conclusion section that ends in the label. To balance the classes, we generate several distinct traces for each positive training example. At inference time, we sample 10 generations per example and estimate the probability of a positive outcome as the fraction of generations whose final answer is `Positive`.

Table 21: Average AUROC and AUPRC of the fine-tuning variants over four EHRSHOT tasks (acute MI, hyperlipidemia, hypertension, and pancreatic cancer) in an earlier version of the project. Some system prompts were still in beta testing in that version, so these numbers are not comparable with the main results. Subscripts denote the confidence intervals reported in that version. **Blue**: best in column. $n = 16$ training sets contain 8 examples per class.

	$n = 16$		$n = \text{All}$	
Method	AUROC	AUPRC	AUROC	AUPRC
SFT-CoT	$.650_{.626-.674}$	$.149_{.125-.176}$	$.744_{.723-.763}$	$.186_{.164-.210}$
NaiveText-LoRA	$.704_{.680-.729}$	$.191_{.160-.224}$	$.756_{.734-.778}$	$.258_{.222-.294}$
Global-Rubric-LoRA	$.695_{.672-.719}$	$.202_{.168-.236}$	$.783_{.763-.803}$	$.296_{.258-.338}$

LoRA variants outperform SFT-CoT. Table 21 reports the average performance of the three fine-tuning variants over the four tasks. Both LoRA variants outperform SFT-CoT at both training-set sizes and in both metrics, with the largest gaps in AUPRC: at $n =$ All, Global-Rubric-LoRA reaches 0.296 and NaiveText-LoRA 0.258, compared to 0.186 for SFT-CoT. At $n = 16$, NaiveText-LoRA and Global-Rubric-LoRA reach 0.704 and 0.695 AUROC, respectively, compared to 0.650. The gap is smallest in AUROC at $n =$ All between NaiveText-LoRA and SFT-CoT (0.756 vs 0.744), where the confidence intervals overlap. The ordering is not uniform across tasks, as SFT-CoT has the highest AUROC of the three on pancreatic cancer at both training-set sizes. The embedding-based Global-Rubric of the earlier version also outperformed SFT-CoT on average at both training-set sizes. LoRA fine-tuning and the embedding approach performed similarly.

H.2 Prompted classification over rubric representations

We also prompt an LLM for a `Yes`/`No` verdict over rubric representations in the setup of the main paper, as for the CoT baselines in Section 4.3, which performs notably below the embedding and logistic regression pipeline. This is consistent with the weaker performance of SFT-CoT in Section H.1, which is fine-tuned to generate a reasoning trace followed by a `Positive`/`Negative` answer, and with Wu et al. [69], who report that chain-of-thought prompting can degrade clinical text understanding. Our interpretation is that a rubric governs which evidence is available and how it is organized, while the calibration of the decision is left to the downstream classifier. With a frozen embedding model, this calibration happens in the logistic regression head, which is fit directly to the labels.

I Evaluation beyond EHRSHOT: Online recruitment fraud detection

To evaluate the pipeline outside healthcare, we apply Global-Rubric-Tabular to EMSCAD [63], which consists of 17,583 online job advertisements, 4.9% of which are fraudulent (10,550/2,637/4,396 training/validation/test examples). Each advertisement is semi-structured text with partially filled categorical fields and inconsistent vocabularies, so the data do not come as a clean feature table. Vidros et al. [63] hand-designed an expert ruleset of 19 features for this dataset, whereas our pipeline produces 360 features, and both feature sets are used with an XGBoost classifier [6]. Several rubric features are absent from the expert ruleset. For instance, a flag for generic job titles fires on 17% of fraudulent and 0.4% of legitimate ads.

Global-Rubric-Tabular improves on the expert ruleset with 400 and 1,000 training examples and with the full training set (Table 22), most strongly in AUPRC, which rises from 0.616 to 0.841 with the full training set. All six differences are significant at $p < 0.001$ under a paired bootstrap over the shared test split.

Table 22: AUROC and AUPRC on the EMSCAD test split, with XGBoost trained on the 19-feature expert ruleset of Vidros et al. [63] or on the 360 features of Global-Rubric-Tabular. n is the total number of training examples.

	$n = 400$		$n = 1{,}000$		$n =$ All	
Features	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
Expert ruleset	.896	.355	.920	.459	.934	.616
Global-Rubric-Tabular	.928	.495	.964	.685	.974	.841

J Probing for data leakage

The pretraining data of the LLMs we use may extend past the release of EHRSHOT, raising the question of whether they have seen its records. We first asked GPT5-Mini and Qwen3-8B to describe the EHRSHOT benchmark, and neither could, while both recognized and described MIMIC-III [22]. A model's inability to describe a dataset does not show that the data are absent from its pretraining corpus, so we also run a likelihood-based membership test on Qwen3-8B, similar to Mattern et al. [33] and Shi et al. [49]. We take 100 blocks of rows (capped at 1,024 tokens) from each of the EHRSHOT files `ehrshot.csv` and `labeled_patients.csv`, and build a perturbed copy of each block by permuting the values within each column across records. Each column thus keeps its set of

values, while the combination of fields within a record is scrambled. We then report the AUROC of separating real blocks from their perturbed copies based on the model's likelihood, where a value well above 0.5 would suggest memorization. As a positive control, we use rows of the iris dataset, which is widely available on the web.

The probe separates the iris rows perfectly, while it stays near chance on both EHRSHOT files (0.57 and 0.52 AUROC; Table 23). While this test does not prove that EHRSHOT was not memorized, it helps build confidence. Note also that our comparisons are less exposed to leakage than absolute scores. NaiveText uses the same embedding backbone as our rubric methods, and the global rubric representation is a transformation of the naive serialization, so any leakage into the backbone would also help NaiveText, which trails the rubric methods. Further, individual-level access to EHRSHOT is gated, and CLMBR-T, pretrained on patients drawn from the same distribution as EHRSHOT, is the most exposed model by construction.

Table 23: Membership probe on Qwen3-8B. AUROC of separating real data blocks from copies with values permuted within columns, based on the model's likelihood. Values near 0.5 indicate no detectable memorization.

Data	AUROC
iris (positive control)	1.00
EHRSHOT `ehrshot.csv`	0.57
EHRSHOT `labeled_patients.csv`	0.52

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1.4 state our contributions (global and local rubrics for representation design) and the evaluation scope. These are matched by the methodology in Section 2 and the empirical results in Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 7 contains a dedicated "Limitations and Future Work" paragraph noting the single-benchmark scope, absence of richer modalities (notes, imaging), and context-window-bounded rubric synthesis.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include theoretical results; the contribution is methodological and empirical (Sections 2 and 5).

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Sections 2 to 4 describe the rubric synthesis/application procedure, downstream classifiers, embedding backbones, and evaluation protocol; the EHRSHOT subsampling procedure is detailed in Section B; full prompts used for every LLM step are provided in Section D; and example learned rubrics are included in Section G.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide code for rubric synthesis/application, embedding-based downstream training, and evaluation in https://github.com/demireal/LRRL. EHRSHOT [68] is publicly available under credentialed access through Stanford as described in Section 3, and our subsampling procedure for the train/val/test splits is documented in Section B together with sample counts in Table 1.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 specifies the downstream classifiers (logistic regression on frozen embeddings; XGBoost/LightGBM for tabular and count baselines); Section 3 and Section B describe data splits, label-balanced sampling, and validation-based hyperparameter selection; embedding backbones are listed in Section 4 with additional results in Section C.5; and the LLMs used for rubric synthesis/application (GPT5-Mini, GPT-5.2) are stated in Section 2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report 95% confidence intervals obtained by bootstrapping the test split. Confidence bands are shown in Figures 8 and 9 in Section C.2, and Table 2, in Section C.1, and the per-task tables in Section C.6.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We discuss the qualitative inference cost of each method ($\mathcal{O}(1)$ vs. $\mathcal{O}(N)$ LLM calls; CPU-only deployment for Global-Rubric-Auto/Tabular) in Sections 1.4 and 7. We also report further API and compute costs in Section E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We use only the publicly released, deidentified EHRSHOT benchmark [68] described in Section 3, and access GPT models exclusively through a HIPAA-compliant Microsoft Azure OpenAI Service deployment (see footnote in Section 1.4).

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Section 7 discusses some positive and negative impacts as well as limitations.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not release any new pretrained models, generators, or scraped datasets; we release task-specific rubrics and downstream classifiers fitted on the publicly released EHRSHOT benchmark, which does not pose dual-use or generative-model misuse risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets (EHRSHOT [68]), models (CLMBR-T [68], Qwen3-Embedding-8B [77], Qwen3-8B [43], Llama-3-8B [36], Mistral-7B [19], GPT-5 family [39], OpenAI text-embedding-3-large [38]), and tools (LightGBM [23], XGBoost [6], LLM2Vec [5]) are cited at point of use in Sections C.5, 4, 4.2 and 4.3, and used under their respective licenses and terms.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We release the prompts used to produce the rubrics in Section D and example rubrics are shown in Section G (Sections G.1 and G.2), with feature-schema statistics described in the case study referenced from Section 6.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This work does not involve crowdsourcing or new human-subjects data collection; we use the publicly released, deidentified EHRSHOT benchmark [68] (Section 3).

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: We do not perform new research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs are central to our method: Section 2 describes how GPT5-Mini synthesizes global rubrics, applies them, and produces local rubrics; GPT-5.2 generates the rubric-application parser and tabularization scripts. Models, prompts, and roles for each step are documented in Sections 2, 4 and 4.3 and Section D.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.