COMPLEMENTARY EVIDENCE REASONING OVER STRUCTURED AND UNSTRUCTURED ELECTRONIC HEALTH RECORDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Structured and unstructured features derived from the same record often overlap, so learning from both requires distinguishing complementary evidence from redundant information. In Electronic Health Records (EHRs), clinical notes capture key observations that are not readily standardized into medical codes, such as symptom severity and treatment response, while much of their remaining content is already in the structured record. Existing EHR methods encode or summarize notes before fusing them with the structured record, without asking what the note adds beyond the codes. Reasoning-enhanced methods reason to represent an input on its own terms, without feedback from the downstream task. In both cases, reasoning is not trained to select complementary evidence for predictive utility. To this end, we propose **CERE** (Complementary Evidence Reasoning for EHRs), which uses LLM reasoning to learn task-aligned representations of clinical notes that complement structured records. CERE first maps visit codes to soft prompts that tell the LLM what the structured record already captures. Guided by these prompts, the LLM reasons about what the note adds and encodes that reasoning as a dense representation trained for prediction. Finally, reinforcement learning refines the reasoning itself, rewarding evidence selections that lower prediction loss. Experiments on MIMIC-III and MIMIC-IV show that CERE consistently outperforms strong baselines across multiple clinical prediction tasks.

1 INTRODUCTION

Structured features and unstructured text associated with the same sample can contain both overlapping and complementary information (Liang et al., 2023; Dufumier et al., 2025; Yang et al., 2025). This motivates identifying what the text contributes to prediction beyond the structured information already available. Electronic Health Records (EHRs) provide an important setting for this problem. Structured diagnoses, procedures, and medication codes standardize clinical events and support longitudinal modeling across patients and visits (Choi et al., 2016; Jiang et al., 2024), while clinical notes describe patient-specific observations, such as symptom severity and treatment response, that are not readily standardized into codes but carry critical predictive signal (Tayefi et al., 2021). At the same time, much of a note restates what the codes already record, such as the diagnoses themselves (Seinen et al., 2025). Effective reasoning over clinical notes must therefore account for what the structured record already captures and identify additional evidence useful for prediction.

Existing approaches combine clinical text with structured features through representation fusion (Zhang et al., 2020a) or joint note-code summarization and soft prompting (H Moghaddam et al., 2026). These approaches integrate the two sources but do not explicitly optimize generated note reasoning for its added predictive value given the structured record. This motivates complementary reasoning: determining which evidence in unstructured text remains informative given the structured context. Large Language Models can generate intermediate reasoning steps (Wei et al., 2022), and recent work shows that reasoning before representation extraction improves dense representations (Yan et al., 2025; Cui et al., 2026; Sun et al., 2026). Together, these suggest a design: reason over the note with the structured record as context, and utilize the outcome of that reasoning for the representation learning rather than the note itself.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

The central challenge is to guide complementary reasoning toward evidence that improves downstream prediction. Reasoning-enhanced embedding methods optimize the representation extracted after reasoning (Yan et al., 2025; Cui et al., 2026), or train the reasoning itself with contrastive rewards (Sun et al., 2026). However, these methods do not use a joint structured-text prediction loss to guide the selection of complementary textual evidence. Consequently, the LLM may produce plausible reasoning that adds little predictive value. Reinforcement learning closes this gap by refining evidence selection through task feedback (Yadav et al., 2021), rewarding complementary reasoning that contributes predictive information beyond the structured context.

We introduce **CERE** (**C**omplementary **E**vidence **R**easoning for **E**HRs), which uses LLM reasoning to learn task-aligned representations of clinical notes that complement the structured record. CERE first maps visit codes to continuous soft prompts (Lester et al., 2021; Wu et al., 2023; Vu et al., 2021; H Moghaddam et al., 2026), trained with a discriminative reconstruction objective so that each prompt tells the LLM what its own visit already captures. Guided by these prompts, the LLM reasons about what the note adds and outputs a dense representation, which is fused with the structured record and adapted through the prediction loss. Finally, group-relative reinforcement learning (Shao et al., 2024; Guo et al., 2025) refines the reasoning itself, using a frozen downstream predictor as the judge and rewarding trajectories that lower prediction loss. We evaluate CERE on MIMIC-III and MIMIC-IV (Johnson et al., 2016; 2023) for hospital readmission prediction and medication recommendation, against structured-record baselines, LLM-based EHR models, and fused note encoders. CERE achieves the best performance across both datasets.

2 METHODOLOGY

2.1 PROBLEM SETUP AND LEARNING OBJECTIVE

For each patient i, we observe an ordered sequence of historical visits $\{V_{i,t}\}_{t=1}^{T_i}$, where T_i is the number of visits available before the target visit. Each visit contains structured diagnosis, procedure, and medication codes, $V_{i,t} = \{x_{i,j,t}\}_{j=1}^{|V_{i,t}|}$, with $V_{i,t}^{(s)} \subseteq V_{i,t}$ denoting codes of type $s \in \{d, p, m\}$. Each visit is also accompanied by a clinical note $S_{i,t} = \{w_{i,t,\ell}\}_{\ell=1}^{|S_{i,t}|}$.

The prediction target is

$$Y_i \in \begin{cases} \{0,1\}, & \text{readmission prediction,} \\ \{0,1\}^K, & \text{medication recommendation,} \end{cases} \tag{1}$$

where K is the medication vocabulary size and, for medication recommendation, $Y_{i,k} = 1$ indicates that medication k is prescribed at the target visit $T_i + 1$. The framework is otherwise shared across the two tasks. Only the output space, prediction head, and task loss differ.

Given these two modalities, our objective is not to learn a generic representation of the clinical note, but to capture the information in the note that remains useful after accounting for the structured record. We formalize this target as the conditional residual representation:

$$\boldsymbol{z}_{i,t}^{\text{res}} = \mathcal{F}\big(S_{i,t} \mid V_{i,t}\big), \tag{2}$$

where $\mathcal{F}$ is trained to produce note representations that improve prediction of Y_i given $V_{i,t}$. Structured conditioning encourages the use of complementary note evidence (Seinen et al., 2025; Bayramli et al., 2022). Here, "residual" denotes this conditional predictive role.

2.2 METHODOLOGY OVERVIEW

Figure 1 presents the architecture of **CERE** and its three optimization stages. For each visit, structured codes are mapped to continuous soft prompts that condition LLM reasoning over the corresponding clinical note. This reasoning seeks evidence that adds predictive information beyond the codes. The hidden state at the `<embed>` token provides a residual embedding, which is combined with code representations for patient-level prediction.

Multi-Stage Optimization: Three stages realize Equation 2. First, **structured context grounding** learns soft prompts that represent the coded visit through a discriminative reconstruction objective. Second, **residual evidence learning** trains the predictor, then jointly optimizes the predictor and LoRA adapters to extract complementary note evidence and encode it for prediction. Third,

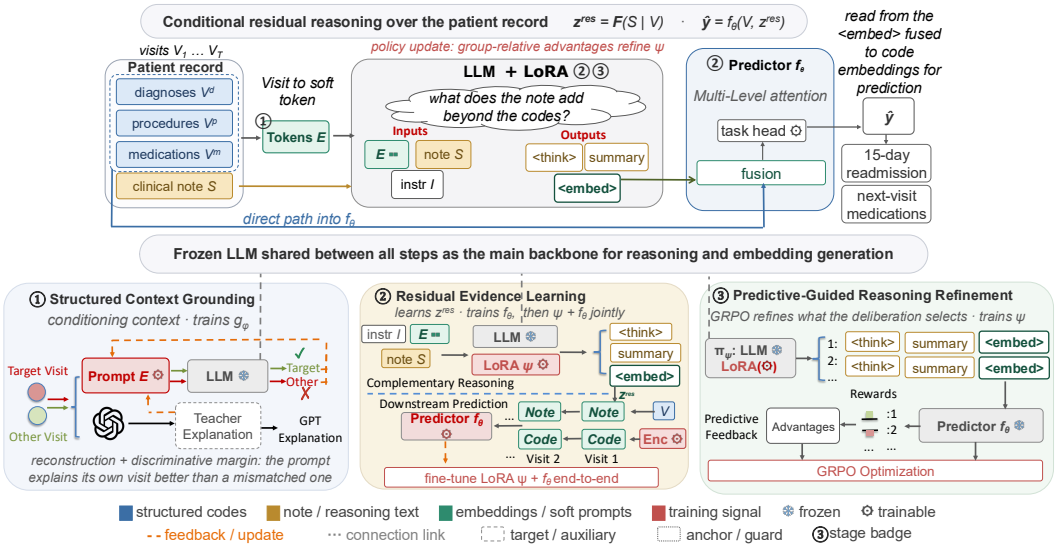


Figure 1: **Top:** CERE architecture. Learned soft prompts encode structured visit codes and condition LLM reasoning over clinical notes. The <embed> hidden state provides a residual embedding, which is combined with code representations for patient-level prediction. **Bottom:** Three optimization stages: grounding learns visit-specific prompts through discriminative reconstruction; residual evidence learning optimizes complementary note representations for prediction; and predictor-guided refinement uses feedback from the frozen predictor to improve evidence selection. Numbered badges identify the components optimized at each stage.

predictor-guided reasoning refinement treats reasoning generation as a policy and uses feedback from the frozen predictor to refine evidence selection. The backbone LLM $\mathcal{M}$ remains frozen throughout, while the stages update the soft prompts, LoRA adapters, and predictor as indicated.

2.3 Structured Context Grounding (Stage 1)

To determine what the clinical note adds beyond the coded record, the LLM must first know what information is already present in the structured modality. A straightforward approach is to convert the codes into text. However, textualizing the codes consumes additional context tokens and can make distinctions between diagnosis, procedure, and medication codes less explicit to the LLM (H Moghaddam et al., 2026). Instead, we map the structured visit directly into a small set of continuous soft-prompt vectors in the LLM input space (Lester et al., 2021; Li & Liang, 2021). These vectors provide the LLM with a compact representation of the coded visit that can directly condition its reasoning over the note.

To faithfully encode each visit, soft prompts are trained to let the frozen LLM reconstruct a teacher-generated description of its structured codes. The teacher organizes the same code-supported information into coherent visit-level text rather than flat ontology names to align the prompts with the LLM's representation space. Because reconstruction alone may favor generic prompts that support many plausible descriptions, we add a discriminative objective that requires each prompt to explain its own visit better than a mismatched visit from another patient.

To construct such a context, we use a code encoder g_ϕ, parameterized by ϕ, that maps the three code types to P soft-prompt vectors,

$$\boldsymbol{E}_{i,t} = g_\phi\big(V_{i,t}^{(d)}, V_{i,t}^{(p)}, V_{i,t}^{(m)}\big) \in \mathbb{R}^{P\times D}. \tag{3}$$

Let $\mathcal{T}_{i,t}$ denote a natural-language description of the structured codes in $V_{i,t}$ generated by a teacher LLM (GPT-5.4 mini) from code identifiers and their official descriptions. The teacher has no access to the clinical note or prediction label. Let $\mathrm{CE}(\mathcal{T} \mid \boldsymbol{E})$ denote the autoregressive cross-entropy assigned by the frozen LLM when conditioned on $\boldsymbol{E}$. For a mismatched visit $V_{i',t'}$ from another

patient, we minimize

$$\mathcal{L}_{\text{align}} = \text{CE}\big(\mathcal{T}_{i,t} \mid \boldsymbol{E}_{i,t}\big) + \lambda \,\text{softplus}\Big(\text{CE}\big(\mathcal{T}_{i,t} \mid \boldsymbol{E}_{i,t}\big) - \text{CE}\big(\mathcal{T}_{i',t'} \mid \boldsymbol{E}_{i,t}\big)\Big). \tag{4}$$

The first term encourages faithfulness by requiring the soft prompt to support reconstruction of the teacher-generated description of its own structured visit. The second promotes visit specificity by encouraging lower reconstruction loss for the matched description than for a mismatched description. The softplus provides a smooth positive-part penalty. Only ϕ is optimized; all parameters of $\mathcal{M}$ remain frozen, so grounding acts as an input-space bridge that preserves the language and reasoning capabilities of the backbone. Detailed encoder architecture is provided in Appendix A.1.1, and the construction of $\mathcal{T}_{i,t}$ and the corresponding teacher prompt are provided in Appendix A.2.1.

2.4 Residual Evidence Learning (Stage 2)

Clinical notes contain both patient-specific information absent from structured EHR variables (Feng et al., 2020) and information that overlaps with the coded record (Seinen et al., 2025). We therefore condition note reasoning on structured codes to encourage the use of complementary evidence targeted by Equation 2. Before representation extraction, the LLM reasons about what the note adds to the structured record. This reasoning provides additional context for the representation.

For each visit, $\mathcal{M}$ receives the soft prompt $\boldsymbol{E}_{i,t}$, note $S_{i,t}$, and an instruction $\mathcal{I}$:

$$R_{i,t} = \mathcal{M}_{\text{greedy}}\big(\boldsymbol{E}_{i,t}, S_{i,t}, \mathcal{I}\big), \qquad \big|R_{i,t}^{\text{think}}\big| \le B, \tag{5}$$

where $R_{i,t}$ is the generated reasoning trajectory and B bounds its deliberation length. Each trajectory is generated once and cached during this stage. The complete instruction and output format are provided in Appendix A.2.2.

Following decoder-only embedding approaches (Li et al., 2025), we append a dedicated $\langle\text{embed}\rangle$ token after the generated context and use its final-layer hidden state:

$$\boldsymbol{z}_{i,t} = \boldsymbol{h}^{(L)}_{\langle\text{embed}\rangle}\big(\boldsymbol{E}_{i,t}, S_{i,t}, \mathcal{I}, R_{i,t}\big) \in \mathbb{R}^D. \tag{6}$$

The residual representation is fused with structured visit features and consumed by a reverse-time attention predictor based on RETAIN (Choi et al., 2016). For each structured stream $s \in \{d, p, m\}$, we concatenate the structured features with the projected residual representation, then apply a linear transformation and layer normalization: $\tilde{\boldsymbol{e}}^{(s)}_{i,t} = \text{LN}\Big(\boldsymbol{W}_f^{(s)}[\boldsymbol{e}^{(s)}_{i,t}; \boldsymbol{W}_z \boldsymbol{z}_{i,t}]\Big)$. This fusion allows the predictor to use complementary evidence from clinical notes alongside structured features. The resulting visit sequences are pooled by separate reverse-time attention streams and passed to a task-specific prediction head. We optimize binary cross-entropy for readmission prediction and the sum of label-wise binary cross-entropies for medication recommendation, as detailed in Appendix A.1.4. The complete predictor formulation is provided in Appendix A.1.2.

Although the reasoning trajectory may identify useful complementary evidence, the resulting LLM representation is not necessarily optimized for the downstream prediction task. We therefore adapt the LLM representation using the prediction loss. The generated reasoning trajectories are kept fixed, and $\mathcal{L}_{\text{pred}}$ is backpropagated through the representation readout into low-rank adapters (Hu et al., 2021) with parameters ψ (see Appendix A.1.3):

$$\min_{\theta,\psi} \; \mathbb{E}\left[\mathcal{L}_{\text{pred}}\big(f_\theta(V_i, \boldsymbol{z}_i(\psi)), Y_i\big)\right]. \tag{7}$$

The LLM base weights, code encoder g_ϕ, and cached reasoning trajectories remain frozen.

2.5 Predictive-Guided Reasoning Refinement (Stage 3)

The remaining challenge is to optimize which evidence the LLM selects during reasoning. In Stage 2, the generated reasoning trajectories are fixed, so the prediction loss can improve how the selected evidence is represented but cannot directly change which reasoning tokens are generated. Because token generation involves discrete sampling, the downstream loss cannot be backpropagated directly through these generation decisions (Schulman et al., 2015). We therefore use policy-gradient optimization, which allows the generated reasoning trajectories to be optimized using downstream rewards (Yu et al., 2017). Specifically, we treat reasoning generation as a policy and use the trained predictor to evaluate the resulting representations.

Prediction difficulty varies across patients, so comparing absolute rewards across patients can confound trajectory quality with patient-specific predictability. We therefore compare multiple reasoning trajectories for the same patient and normalize their rewards within the group, following the group-relative formulation of GRPO (Shao et al., 2024).

The policy π_ψ is initialized from the Stage-2 checkpoint, and the trained predictor is frozen as the judge. For each patient, we sample G patient-level rollouts. Each rollout g resamples one reasoning trajectory for every historical visit $t = 1, \ldots, T_i$ and regenerates the corresponding residual representations $\{\boldsymbol{z}_{i,t}^{(g)}\}_{t=1}^{T_i}$. Let $\mathcal{L}_{\text{aug}}^{(g)}$ denote the prediction loss obtained by the frozen predictor from the complete set of representations generated by rollout g. We denote by $r_{\text{fmt}}^{(g)}$ the average format score across the visit-level reasoning trajectories in rollout g; its scoring rule is given in Appendix A.2.3. We define

$$r^{(g)} = \alpha\, r_{\text{fmt}}^{(g)} - \gamma\, \mathcal{L}_{\text{aug}}^{(g)} - \delta \max\bigl(0, d^{(g)} - \tau_d\bigr). \tag{8}$$

The coefficients α, γ, and δ weight format quality, prediction loss, and representation drift, respectively. Within each patient group, the prediction component favors trajectories with lower loss under the frozen predictor. Structured conditioning and the reasoning instruction guide evidence selection toward information that complements the coded record. The final term penalizes representation drift $d^{(g)}$ beyond tolerance τ_d. Generation and representation extraction share the adapters ψ, so policy updates can change both the reasoning trajectories and their embeddings. The drift term penalizes rollouts whose mean relative representation distance exceeds τ_d. We measure this quantity using relative representation drift:

$$d_{i,t}^{(g)} = \frac{\left\| \boldsymbol{z}_{i,t}^{(g)} - \boldsymbol{z}_{i,t}^{\text{ref}} \right\|_2}{\left\| \boldsymbol{z}_{i,t}^{\text{ref}} \right\|_2 + \varepsilon_d}, \qquad d^{(g)} = \frac{1}{T_i} \sum_{t=1}^{T_i} d_{i,t}^{(g)}, \tag{9}$$

where $\boldsymbol{z}_{i,t}^{\text{ref}}$ is the cached pre-refinement representation and ε_d ensures numerical stability. Relative L2 drift captures changes in both magnitude and direction, as either can affect the frozen predictor after projection and fusion. Normalization by the reference norm makes the measure relative to each visit's representation scale. The penalty in Equation 8 is activated only beyond tolerance τ_d. Within each patient group, rewards are normalized as

$$A^{(g)} = \frac{r^{(g)} - \mu_r}{\max(\sigma_r, \sigma_{\min})}, \tag{10}$$

where μ_r and σ_r are the group mean and standard deviation, and $\sigma_{\min} = 0.05$ prevents unstable normalization when the sampled trajectories receive nearly identical rewards.

Following the group-relative formulation of GRPO (Shao et al., 2024), we optimize

$$\mathcal{L}_{\text{GRPO}} = -\mathbb{E}_g \left[\frac{1}{|R^{(g)}|} \sum_{\ell=1}^{|R^{(g)}|} \rho_\ell^{(g)} A^{(g)} \right] + \beta \mathcal{L}_{\text{ref}}, \tag{11}$$

where $R^{(g)}$ collects the generated tokens across all visits in rollout g, and $\rho_\ell^{(g)}$ is the token-level likelihood ratio between the current policy π_ψ and the rollout policy π_{old}. We take one optimizer step per fresh rollout group, with advantages and old log-probabilities detached.

The reference penalty $\mathcal{L}_{\text{ref}}$ averages $\exp(u) - u - 1$ over sampled tokens, where $u = \log \pi_{\text{ref}} - \log \pi_\psi$ and π_{ref} is the frozen Stage-2 policy. Reference log-probabilities are detached. This nonnegative penalty discourages discrepancies between current and reference token probabilities. Only the Stage-3 adapters ψ are optimized; the predictor, code encoder, and LLM base weights remain frozen. The exact penalty and rollout mechanics are provided in Appendix A.3.

2.6 Sequential Optimization

The objectives are optimized sequentially:

$$\min_{\phi} \mathcal{L}_{\text{align}} \longrightarrow \min_{\theta, \psi} \mathcal{L}_{\text{pred}} \longrightarrow \min_{\psi} \mathcal{L}_{\text{GRPO}}. \tag{12}$$

The ordering follows the dependencies between the components. Structured grounding must first establish a faithful and visit-specific representation of the coded record before it can condition note reasoning. The downstream predictor and representation readout must then learn how selected evidence supports the target task before the predictor can serve as a useful judge for policy refinement.

Table 1: Statistics of the EHR datasets. All statistics are computed over the full cohort of each dataset, with the training and test sets combined.

	MIMIC-III	MIMIC-IV
# of instances	12,456	49,424
Avg. visits per instance	2.19	5.49
Avg. unique diagnoses per patient	21.20	37.05
Avg. unique procedures per patient	6.10	5.45
Avg. unique drugs per patient	58.19	59.03
# of Medication labels	3,565	4,811
readmission rate per instance	8.73%	18.54%

We use AdamW (Loshchilov & Hutter, 2017) for gradient-based optimization. After policy refinement, we regenerate the residual representations and retrain the downstream predictor from scratch with $\mathcal{L}_{\text{pred}}$. The frozen Stage-2 predictor provides a fixed reward during policy optimization, while the final refit adapts the downstream predictor to the refined representations. Algorithm 1 in Appendix A.4 summarizes the procedure.

3 Evaluation

3.1 Datasets

In this study, we employ two real-world, publicly available EHR datasets, and the statistics are available in Table 1:

MIMIC-III: (Johnson et al., 2016) contains de-identified records of patients admitted to the intensive care units of the Beth Israel Deaconess Medical Center between 2001 and 2012, including structured diagnosis, procedure, and medication codes together with clinical notes.
MIMIC-IV: (Johnson et al., 2023) covers admissions to the same medical center between 2008 and 2019 and provides the same combination of structured codes and clinical documentation over a larger and more recent patient population. For computational feasibility, we uniformly sample 20% of eligible MIMIC-IV patients before the train/test split. All methods use the same sampled cohort and matched patient-level splits.

3.2 Implementation Details

We evaluate CERE on 15-day readmission prediction and next-visit medication recommendation. Each prediction instance uses an admission and the patient's preceding history to predict readmission within 15 days or the medications prescribed at the next admission. Single-admission patients are excluded. We evaluate all methods over six random seeds with patient-level 80/20 train/test splits shared across methods for each seed, and report the mean and 95% confidence interval. We use the post-trained instruction-following Qwen3.5-4B model as the backbone. We use implementations from PyHealth (Yang et al., 2023). Implementation and hyperparameter details appear in Appendices A.5 and A.6. The implementation is publicly available.[1]

3.3 Baselines

We compare CERE against a diverse set of established EHR prediction models. The deep learning baselines include RETAIN (Choi et al., 2016), Deepr (Nguyen et al., 2017), GRAM (Choi et al., 2017), a standard Transformer (Vaswani et al., 2017), AdaCare (Ma et al., 2020), StageNet (Gao et al., 2020), GRASP (Zhang et al., 2021), ADORE (Cheong et al., 2023), and ARCI (H Moghaddam et al., 2024). We further compare against recent LLM-based EHR methods, including GraphCare (Jiang et al., 2024), LINKO (Nayebi Kerdabadi et al., 2025), and RePrompT (H Moghaddam et al., 2026). We additionally construct frozen note-encoder baselines fused with code embedding for RETAIN and BERT (Devlin et al., 2019), ModernBERT (Warner et al., 2025), BioClinical-MBERT (Sounack et al., 2025), and Qwen3-Embedding-4B (Zhang et al., 2025) as fixed text encoders. Detailed baseline descriptions are provided in Appendix A.7

[1] `https://anonymous.4open.science/r/ehr-reasoning`.

Table 2: Baseline performance comparison on MIMIC-III and MIMIC-IV. Results are reported as mean ± 95% confidence interval.

	Model	Readmission		Medication	
		AUROC	PRAUC	PRAUC	F1
MIMIC-III	**DL-based Methods**				
	RETAIN (Choi et al., 2016)	70.20 ± 0.91	33.83 ± 0.79	32.97 ± 0.27	18.88 ± 0.50
	Deepr (Nguyen et al., 2017)	70.66 ± 0.89	34.25 ± 0.82	31.17 ± 0.30	17.52 ± 1.13
	GRAM (Choi et al., 2017)	71.75 ± 0.74	34.60 ± 1.15	29.52 ± 0.77	15.59 ± 1.13
	Transformer (Vaswani et al., 2017)	69.79 ± 0.89	33.16 ± 1.14	28.67 ± 0.26	14.44 ± 0.94
	AdaCare (Ma et al., 2020)	71.05 ± 0.87	34.86 ± 0.98	32.03 ± 0.33	18.50 ± 0.53
	StageNet (Gao et al., 2020)	68.74 ± 1.26	32.72 ± 1.03	32.96 ± 0.90	18.15 ± 1.27
	GRASP (Zhang et al., 2021)	68.56 ± 0.63	33.36 ± 0.11	30.22 ± 0.29	16.77 ± 1.07
	ADORE (Cheong et al., 2023)	67.88 ± 1.06	32.42 ± 0.87	29.65 ± 0.63	15.47 ± 0.66
	ARCI (H Moghaddam et al., 2024)	70.53 ± 1.36	34.13 ± 0.94	32.36 ± 0.20	18.75 ± 1.18
	LLM-based Methods				
	GraphCare (Jiang et al., 2024)	71.22 ± 0.85	34.41 ± 0.86	31.12 ± 0.83	17.16 ± 0.90
	LINKO (Nayebi Kerdabadi et al., 2025)	70.73 ± 0.94	34.06 ± 0.88	31.58 ± 0.83	17.66 ± 1.07
	RePrompT (H Moghaddam et al., 2026)	71.84 ± 0.71	34.91 ± 0.76	32.55 ± 0.36	18.45 ± 0.78
	RETAIN + Frozen Note Encoder				
	BERT (Devlin et al., 2019)	72.39 ± 0.75	35.25 ± 0.91	34.19 ± 0.30	19.46 ± 0.83
	ModernBERT (Warner et al., 2025)	72.78 ± 0.68	35.19 ± 0.83	34.07 ± 0.38	19.77 ± 1.04
	BioClinical-MBERT (Sounack et al., 2025)	72.38 ± 0.75	35.94 ± 0.78	34.18 ± 0.34	20.13 ± 0.75
	Qwen3-Embedding-4B (Zhang et al., 2025)	73.23 ± 0.89	36.56 ± 0.94	34.41 ± 0.32	20.63 ± 0.92
	CERE	**75.13** ± 0.79	**38.43** ± 0.71	**36.33** ± 0.55	**22.48** ± 0.83
MIMIC-IV	**DL-based Methods**				
	RETAIN (Choi et al., 2016)	64.22 ± 0.53	33.45 ± 0.71	32.13 ± 0.27	16.70 ± 0.47
	Deepr (Nguyen et al., 2017)	63.75 ± 0.68	32.70 ± 1.31	30.27 ± 0.26	15.60 ± 0.75
	GRAM (Choi et al., 2017)	64.70 ± 0.82	34.56 ± 0.91	29.41 ± 0.70	14.65 ± 1.20
	Transformer (Vaswani et al., 2017)	62.36 ± 0.92	30.82 ± 1.16	30.67 ± 0.20	15.08 ± 0.47
	AdaCare (Ma et al., 2020)	61.43 ± 0.94	29.89 ± 0.85	31.94 ± 0.29	16.54 ± 0.83
	StageNet (Gao et al., 2020)	61.67 ± 1.01	30.64 ± 1.12	33.16 ± 0.25	18.75 ± 1.19
	GRASP (Zhang et al., 2021)	62.37 ± 0.96	31.07 ± 0.72	29.31 ± 0.32	14.17 ± 0.92
	ADORE (Cheong et al., 2023)	61.20 ± 1.06	31.10 ± 1.03	31.09 ± 0.28	17.12 ± 0.79
	ARCI (H Moghaddam et al., 2024)	63.97 ± 0.86	34.25 ± 0.89	34.16 ± 0.49	19.55 ± 0.81
	LLM-based Methods				
	GraphCare (Jiang et al., 2024)	64.31 ± 1.05	34.51 ± 0.94	31.44 ± 0.58	18.22 ± 0.92
	LINKO (Nayebi Kerdabadi et al., 2025)	64.76 ± 0.95	35.04 ± 0.92	31.99 ± 0.58	18.51 ± 0.92
	RePrompT (H Moghaddam et al., 2026)	65.18 ± 0.78	35.58 ± 0.97	32.70 ± 0.48	18.43 ± 0.34
	RETAIN + Frozen Note Encoder				
	BERT (Devlin et al., 2019)	64.83 ± 0.45	34.90 ± 0.99	37.14 ± 0.21	21.39 ± 0.97
	ModernBERT (Warner et al., 2025)	65.26 ± 0.58	35.44 ± 0.81	37.11 ± 0.27	20.90 ± 0.58
	BioClinical-MBERT (Sounack et al., 2025)	65.99 ± 0.33	36.22 ± 0.68	37.15 ± 0.29	21.49 ± 0.64
	Qwen3-Embedding-4B (Zhang et al., 2025)	66.55 ± 0.48	37.55 ± 0.71	38.42 ± 0.25	23.13 ± 0.96
	CERE	**68.71** ± 0.68	**39.43** ± 0.85	**39.54** ± 0.55	**25.11** ± 0.83

3.4 Comparison with Baselines

Deep Learning Methods. CERE achieves the highest mean performance across both datasets and tasks in Table 2. The strongest structured baseline varies with the prediction target: GRAM is competitive for readmission, while RETAIN, StageNet, and ARCI perform well for medication recommendation. Attention and medical ontologies help these models capture temporal and semantic relationships among codes, but they cannot directly access details recorded only in clinical notes. CERE incorporates this additional information into the patient representation.

LLM Methods. GraphCare constructs patient graphs with external medical knowledge, LINKO learns concept representations across medical ontologies, and RePrompT combines joint note and code summaries with recurrent and structured-encoder prompts. These approaches improve the

Table 3: Ablation study on two datasets, with results reported as mean ± 95% confidence interval.

	Model Variant	Readmission		Medication	
		AUROC	PRAUC	PRAUC	F1
MIMIC-III	w/o Conditional Encoding	73.05 ± 0.91	37.21 ± 0.64	34.32 ± 0.61	19.11 ± 0.80
	w/o Grounding CL	73.82 ± 0.81	37.49 ± 0.68	34.75 ± 0.49	19.47 ± 0.82
	w/o Predictive LoRA Adaptation	74.81 ± 0.86	38.28 ± 0.71	35.65 ± 0.58	21.35 ± 0.76
	w/o Residual Reasoning	74.16 ± 0.97	37.53 ± 0.59	35.24 ± 0.52	20.43 ± 0.70
	w/o RL Refinement	74.62 ± 0.72	38.09 ± 0.55	35.84 ± 0.46	21.97 ± 0.65
	CERE	**75.13** ± 0.79	**38.43** ± 0.71	**36.33** ± 0.55	**22.48** ± 0.83
MIMIC-IV	w/o Conditional Encoding	66.13 ± 0.78	37.90 ± 0.95	37.92 ± 0.58	22.74 ± 0.87
	w/o Grounding CL	66.92 ± 0.73	38.42 ± 0.86	38.43 ± 0.61	23.56 ± 0.89
	w/o Predictive LoRA Adaptation	68.41 ± 0.83	38.81 ± 0.77	39.05 ± 0.49	24.56 ± 0.74
	w/o Residual Reasoning	67.89 ± 0.64	38.48 ± 0.94	38.73 ± 0.48	24.23 ± 0.70
	w/o RL Refinement	68.08 ± 0.60	38.87 ± 0.82	38.75 ± 0.59	23.96 ± 0.77
	CERE	**68.71** ± 0.68	**39.43** ± 0.85	**39.54** ± 0.55	**25.11** ± 0.83

representation of medical concepts and patient histories, yet they do not consistently outperform the structured baselines in our experiments. CERE focuses on the contribution of clinical notes given the available codes. Structured conditioning targets complementary evidence, while predictive feedback improves joint prediction.

Frozen Note Encoders. We use four text encoders as frozen note encoders and fuse their embeddings with structured code embeddings in a RETAIN predictor. Every encoder improves over RETAIN alone. The gains are particularly pronounced for medication recommendation on MIMIC-IV, where even fixed text representations provide substantial predictive value. Qwen3-Embedding-4B performs best among these baselines, but CERE retains an advantage across all metrics. Thus, the results favor our full approach over the addition of fixed note embeddings. The performance gains might be due to enhanced code-conditioned reasoning and task-specific representation learning.

3.5 Ablation Studies

Table 3 compares CERE with variants that each omit one component. Without conditional encoding, notes are interpreted without structured-code prompts; without grounding CL, prompt alignment loses the contrastive signal that distinguishes matched from mismatched visit contexts. These removals cause the largest declines in mean performance, suggesting that both structured context and its alignment with the LLM matter for note interpretation. Without residual reasoning, the `<embed>` token directly follows the code-conditioned note, with no generated reasoning tokens. Structured conditioning, predictive LoRA, and the predictor are retained. Its lower mean performance supports explicit reasoning before representation extraction under the evaluated configuration. Without predictive LoRA adaptation, the representation encoder is no longer adapted through prediction loss; without RL refinement (details are provided in Appendix A.5), evidence selection receives no reward-based refinement from prediction outcomes. Both removals cause smaller declines, with their relative effects varying across tasks and metrics. Thus, their smaller gains indicate additional benefits from task-specific adaptation, rather than unnecessary components. The ablation of the predictor selector is available in Appendix A.8

3.6 Case Study

Figure 2 compares rationales generated for the same dialysis patient before and after GRPO, with identical greedy decoding budgets. Before optimization, the model focuses on the hospital course and ambiguous creatinine values, then ends without a conclusion. After optimization, it prioritizes missed dialysis, medication non-adherence, and limited access to care and social support. This example illustrates a change in the prioritization of generated evidence after policy refinement. Appendix A.9 presents one additional example.

Visit: hypertensive emergency in a dialysis patient (MIMIC-III).

Before	**After (GRPO)**
Deliberation: "Trends: Creatinine improved (4.0 -> 22*? Wait, looking at dates... [**2193-5-23**] vs [**2193-6-7**]). Actually, looking closely at the provided text structure, there are some date anomalies or placeholders..."	*Deliberation:* "Readmission Risk Factors: Non-adherence to dialysis and meds, living alone without support system, lack of insurance/follow-up care."
Summary: "...It documents specific complications including two seizures (hypertensive leucoencephalopathy), post-ictal coma... discharge labs indicate improvement in creatinine levels (from admission values to ~22 mg/dL range noted in the text snippet, likely indicating a typo in the source data provided as '22*' vs '4.0*', assuming relative trend of management)...[5] *(ends mid-list, no conclusion)*	*Summary:* "...it details that the patient lives alone with no family support in the area and has been unable to obtain health insurance for a year, limiting her access to follow-up care. ... the note explicitly identifies non-adherence as a root cause (missing dialysis sessions and medications), which indicates a high likelihood of recurrence... **Conclusion:** The most predictive information missing from the codes is the social vulnerability and medication/dialysis adherence..."

Figure 2: With the same generation budget, GRPO shifts reasoning from measurement details to evidence relevant to readmission. Orange: descriptive drift. Blue: outcome-directed selection.

4 Related Work

Deep Learning Models. RETAIN (Choi et al., 2016) uses reverse-time attention to model patient histories, while GRAM (Choi et al., 2017) and HAP (Zhang et al., 2020b) incorporate medical ontologies to represent related concepts. AdaCare (Ma et al., 2020) captures clinical patterns at multiple temporal scales, and StageNet (Gao et al., 2020) models disease progression through stage-aware representations. HTP-Star (Xu et al., 2024a) uses a hypergraph transformer with pretraining and fine-tuning to accommodate patients with different available features. CERE complements these structured-record models by conditioning note reasoning on coded patient histories and learning residual representations that capture predictive information beyond the structured record.

LLM-Based Models. GraphCare (Jiang et al., 2024) and RAM-EHR (Xu et al., 2024b) use external knowledge for EHR prediction, while LINKO (Nayebi Kerdabadi et al., 2025) combines LLM embeddings with medical ontologies. KARE (Jiang et al., 2025) retrieves knowledge graph community summaries to enrich patient context for LLM-based clinical prediction. Other studies examine clinical prompting (Zhu et al., 2024) and continuous prompts for frozen LLMs (Lester et al., 2021; Li & Liang, 2021). RePrompT (H Moghaddam et al., 2026) connects structured EHR encoders to LLMs through recurrent visit-level states and soft prompts. CERE targets note evidence that adds predictive information beyond the structured record.

Reasoning-Enhanced LLMs. Following chain-of-thought prompting (Wei et al., 2022), O1 Embedder (Yan et al., 2025) and Think-Then-Embed (Cui et al., 2026) generate intermediate reasoning for representation learning. GRACE (Sun et al., 2026) trains rationale generation with contrastive rewards. TRACE (Hao et al., 2026) compresses generated reasoning into a dedicated embedding token and adapts reasoning use to query complexity for multimodal retrieval. Related work studies prediction-supervised evidence selection (Lei et al., 2016) and reinforcement learning for reasoning (Shao et al., 2024; Guo et al., 2025). Unlike these approaches, CERE focuses on extracting complementary information from clinical notes by conditioning reasoning on structured EHR context and optimizing the resulting residual representation for downstream prediction.

5 Conclusion

We introduced CERE to learn complementary representations from overlapping EHR modalities. Discriminatively aligned continuous prompts ground clinical note reasoning in structured records, while predictive optimization shapes dense representations. Group-relative reinforcement learning aligns evidence selection with downstream utility. We evaluated CERE on two datasets for hospital readmission prediction and medication recommendation. CERE frames multimodal representation learning as conditional evidence selection.

6 AI DISCLOSURE

In this work, we used generative AI tools to improve writing clarity, check grammar, and assist with software code writing. These tools were not used for any other research or experimental purposes.

7 REPRODUCIBILITY STATEMENT

Sections 3.1 to 3.2 and Table 1 describe the cohorts, tasks, splits, seeds, and backbone. Sections 2.3 to 2.6 define the three training stages, and Algorithm 1 (Appendix A.4) summarizes the full procedure. Appendices A.1 to A.3 specify the architecture, exact prompts, and rollout details; Appendices A.5 to A.6 (Table 4) give training schedules and all hyperparameters; Appendix A.7 describes the baselines. MIMIC-III and MIMIC-IV are available to credentialed users through PhysioNet. The anonymized implementation is at `https://anonymous.4open.science/r/ehr-reasoning`.

REFERENCES

Ilkin Bayramli, Victor Castro, Yuval Barak-Corren, Emily M Madsen, Matthew K Nock, Jordan W Smoller, and Ben Y Reis. Predictive structured–unstructured interactions in ehr models: a case study of suicide prediction. *NPJ digital medicine*, 5(1):15, 2022.

Chin Wang Cheong, Kejing Yin, William K Cheung, Benjamin CM Fung, and Jonathan Poon. Adaptive integration of categorical and multi-relational ontologies with ehr data for medical concept embedding. *ACM Transactions on Intelligent Systems and Technology*, 14(6):1–20, 2023.

Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. *Advances in neural information processing systems*, 29, 2016.

Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. Gram: graph-based attention model for healthcare representation learning. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 787–795, 2017.

Xuanming Cui, Jianpeng Cheng, Hong-you Chen, Satya Narayan Shukla, Abhijeet Awasthi, Xichen Pan, Chaitanya Ahuja, Shlok Mishra, Taipeng Tian, Qi Guo, et al. Think then embed: Generative context improves multimodal embedding. In *International Conference on Learning Representations*, volume 2026, pp. 2690–2709, 2026.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Benoit Dufumier, Javiera Castillo Navarro, Devis Tuia, and Jean-Philippe Thiran. What to align in multimodal contrastive learning? In *International Conference on Learning Representations*, volume 2025, pp. 5408–5432, 2025.

Jinyue Feng, Chantal Shaib, and Frank Rudzicz. Explainable clinical decision support from text. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 1478–1489, 2020.

Junyi Gao, Cao Xiao, Yasha Wang, Wen Tang, Lucas M Glass, and Jimeng Sun. Stagenet: Stage-aware neural networks for health risk prediction. In *Proceedings of the web conference 2020*, pp. 530–540, 2020.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Arya H Moghaddam, Mohsen Nayebi Kerdabadi, Mei Liu, and Zijun Yao. Contrastive learning on medical intents for sequential prescription recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 748–757, 2024.

Arya H Moghaddam, Drew Ross, Mohsen Nayebi Kerdabadi, Dongjie Wang, and Zijun Yao. Reprompt: Recurrent prompt tuning for integrating structured ehr encoders with large language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 26766–26778, 2026.

Xiangzhao Hao, Shijie Wang, Tianyu Yang, Tianyue Wang, Haiyun Guo, and Jinqiao Wang. Trace: Task-adaptive reasoning and representation learning for universal multimodal retrieval. *arXiv preprint arXiv:2603.02929*, 2026.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Pengcheng Jiang, Cao Danica Xiao, Adam Cross, and Jimeng Sun. Graphcare: Enhancing healthcare predictions with personalized knowledge graphs. In *International Conference on Learning Representations*, volume 2024, pp. 16839–16866, 2024.

Pengcheng Jiang, Cao Danica Xiao, Minhao Jiang, Parminder Bhatia, Taha Kass-Hout, Jimeng Sun, and Jiawei Han. Reasoning-enhanced healthcare predictions with knowledge graph community retrieval. In *International Conference on Learning Representations*, volume 2025, pp. 81785–81830, 2025.

Alistair E. W. Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J. Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, Li-wei H. Lehman, Leo A. Celi, and Roger G Mark. Mimic-iv, a freely accessible electronic health record dataset. *Scientific data*, 10 (1):1, 2023.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):160035, 2016.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 107–117, 2016.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Chaofan Li, MingHao Qin, Shitao Xiao, Jianlyu Chen, Kun Luo, Defu Lian, Yingxia Shao, and Zheng Liu. Making text embedders few-shot learners. In *International Conference on Learning Representations*, volume 2025, pp. 38107–38124, 2025.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)*, pp. 4582–4597, 2021.

Paul Pu Liang, Yun Cheng, Xiang Fan, Chun Kai Ling, Suzanne Nie, Richard Chen, Zihao Deng, Nicholas Allen, Randy Auerbach, Faisal Mahmood, et al. Quantifying & modeling multimodal interactions: An information decomposition framework. *Advances in Neural Information Processing Systems*, 36:27351–27393, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Liantao Ma, Junyi Gao, Yasha Wang, Chaohe Zhang, Jiangtao Wang, Wenjie Ruan, Wen Tang, Xin Gao, and Xinyu Ma. Adacare: Explainable clinical health status representation learning via scale-adaptive feature extraction and recalibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 825–832, 2020.

Mohsen Nayebi Kerdabadi, Arya H Moghaddam, Dongjie Wang, and Zijun Yao. Multi-ontology integration with dual-axis propagation for medical concept representation. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 2201–2211, 2025.

Phuoc Nguyen, Truyen Tran, Nilmini Wickramasinghe, and Svetha Venkatesh. Deepr: a convolutional net for medical records. *IEEE journal of biomedical and health informatics*, 21(1):22–30, 2017.

John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. Gradient estimation using stochastic computation graphs. *Advances in neural information processing systems*, 28, 2015.

Tom M Seinen, Jan A Kors, Erik M Van Mulligen, and Peter R Rijnbeek. Using structured codes and free-text notes to measure information complementarity in electronic health records: feasibility and validation study. *Journal of Medical Internet Research*, 27:e66910, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

T Sounack, J Davis, B Durieux, A Chaffin, TJ Pollard, E Lehman, AEW Johnson, M McDermott, T Naumann, C Lindvall, et al. a state-of-the-art long-context encoder for biomedical and clinical nlp. *arXiv preprint arXiv:2506.10896*, 2025.

Jiashuo Sun, Shixuan Liu, Zhaochen Su, Xianrui Zhong, Pengcheng Jiang, Bowen Jin, Peiran Li, Weijia Shi, and Jiawei Han. Grace: Generative representation learning via contrastive policy optimization. In *International Conference on Learning Representations*, volume 2026, pp. 26222–26244, 2026.

Maryam Tayefi, Phuong Ngo, Taridzo Chomutare, Hercules Dalianis, Elisa Salvi, Andrius Budrionis, and Fred Godtliebsen. Challenges and opportunities beyond structured data in analysis of electronic health records. *Wiley Interdisciplinary Reviews: Computational Statistics*, 13(6):e1549, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou, and Daniel Cer. Spot: Better frozen model adaptation through soft prompt transfer. *arXiv preprint arXiv:2110.07904*, 2021.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, et al. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 2526–2547, 2025.

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

Junda Wu, Tong Yu, Rui Wang, Zhao Song, Ruiyi Zhang, Handong Zhao, Chaochao Lu, Shuai Li, and Ricardo Henao. Infoprompt: Information-theoretic soft prompt tuning for natural language understanding. *Advances in neural information processing systems*, 36:61060–61084, 2023.

Ran Xu, Yiwen Lu, Chang Liu, Yong Chen, Yan Sun, Xiao Hu, Joyce C Ho, and Carl Yang. From basic to extra features: Hypergraph transformer pretrain-then-finetuning for balanced clinical predictions on ehr. *Proceedings of machine learning research*, 248:182, 2024a.

Ran Xu, Wenqi Shi, Yue Yu, Yuchen Zhuang, Bowen Jin, May Dongmei Wang, Joyce Ho, and Carl Yang. Ram-ehr: Retrieval augmentation meets clinical predictions on electronic health records. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 754–765, Bangkok, Thailand, August 2024b. Association for Computational Linguistics.

Shweta Yadav, Deepak Gupta, Asma Ben Abacha, and Dina Demner-Fushman. Reinforcement learning for abstractive question summarization with question-aware semantic rewards. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 249–255, 2021.

Ruiran Yan, Zheng Liu, and Defu Lian. O1 embedder: Let retrievers think before action. *arXiv preprint arXiv:2502.07555*, 2025.

Chaoqi Yang, Zhenbang Wu, Patrick Jiang, Zhen Lin, Junyi Gao, Benjamin P. Danek, and Jimeng Sun. Pyhealth: A deep learning toolkit for healthcare applications. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, pp. 5788–5789, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701030.

Zequn Yang, Hongfa Wang, and Di Hu. Efficient quantification of multimodal interaction at sample level. *arXiv preprint arXiv:2506.17248*, 2025.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Chaohe Zhang, Xin Gao, Liantao Ma, Yasha Wang, Jiangtao Wang, and Wen Tang. Grasp: generic framework for health status representation learning based on incorporating knowledge from similar patients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 715–723, 2021.

Dongdong Zhang, Changchang Yin, Jucheng Zeng, Xiaohui Yuan, and Ping Zhang. Combining structured and unstructured data for predictive models: a deep learning approach. *BMC medical informatics and decision making*, 20(1):280, 2020a.

Muhan Zhang, Christopher R King, Michael Avidan, and Yixin Chen. Hierarchical attention propagation for healthcare representation learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 249–256, 2020b.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Yinghao Zhu, Zixiang Wang, Junyi Gao, Yuning Tong, Jingkun An, Weibin Liao, Ewen M Harrison, Liantao Ma, and Chengwei Pan. Prompting large language models for zero-shot clinical prediction with structured longitudinal electronic health record data. *arXiv preprint arXiv:2402.01713*, 2024.

A APPENDIX

Contents of Appendix

A.1 Architecture and Representation Details **14**
A.1.1 Structured code encoder 14
A.1.2 Downstream prediction interface 15
A.1.3 Low-rank adaptation 15
A.1.4 Prediction Loss 15

A.2 Prompts and Output Format **15**
A.2.1 Stage-1 target construction 15
A.2.2 Reasoning instruction 16
A.2.3 Deliberation-then-summary format 17

A.3 Policy Rollout Details **17**

A.4 Training Algorithm **19**

A.5 Implementation Details **20**

A.6 Hyperparameters **20**

A.7 Baselines **20**

A.8 Effect of the Prediction Architecture **21**

A.9 Additional Case Studies **22**

A.1 ARCHITECTURE AND REPRESENTATION DETAILS

A.1.1 STRUCTURED CODE ENCODER

Each code type $s \in \{d, p, m\}$ is processed by an independent encoder path. Codes are embedded using a type-specific lookup table $\boldsymbol{C}^{(s)}$ with width $d_c = 256$. Index 0 is reserved for padding, index 1 for out-of-vocabulary codes, and at most 64 codes are retained per type per visit.

Each code set is contextualized using a two-layer pre-norm Transformer encoder with two attention heads and feed-forward width 512. Transformer self-attention provides contextualized representations of the input code set (Vaswani et al., 2017). Two learned query vectors $\boldsymbol{Q}^{(s)}$ then pool the encoded codes:

$$\boldsymbol{U}_{i,t}^{(s)} = \mathrm{MHA}\left(\boldsymbol{Q}^{(s)}, \mathrm{TF}^{(s)}\big(\boldsymbol{C}^{(s)}[V_{i,t}^{(s)}]\big)\right) + \boldsymbol{O}^{(s)} \in \mathbb{R}^{2\times d_c}, \tag{13}$$

where $\boldsymbol{O}^{(s)}$ is a learned slot-specific offset. The two query vectors are initialized orthogonally so that the slots begin with distinct pooling directions.

Each pooled slot is mapped into the LLM input dimension by a type-specific two-layer MLP with GELU activation. The six resulting vectors, two per code type, are stacked to form $\boldsymbol{E}_{i,t} \in \mathbb{R}^{P\times D}$ with $P = 6$. Continuous prompt vectors provide a direct input-space interface to a frozen language model (Lester et al., 2021). Prefix-based adaptation similarly conditions language models using continuous vectors rather than textualized inputs (Li & Liang, 2021).

The affine normalization scale of the projected vectors is initialized so that the soft-token norm matches the scale of the backbone's input embeddings and remains trainable.

A.1.2 DOWNSTREAM PREDICTION INTERFACE

The longitudinal predictor follows RETAIN's reverse-time attention mechanism (Choi et al., 2016). Each code type has an independent longitudinal stream. Visit-level structured features $\boldsymbol{e}_{i,t}^{(s)} \in \mathbb{R}^{d_e}$, with $d_e = 256$, are formed by summing learned code embeddings within each visit.

The residual representation is projected with the shared map $\boldsymbol{W}_z$ and fused with the structured representation. Two recurrent networks then process each fused stream in reverse chronological order:

$$\boldsymbol{m}_{i,t}^{(s)} = \mathrm{GRU}_a^{(s)}\big(\tilde{\boldsymbol{e}}_{i,T_i}^{(s)}, \ldots, \tilde{\boldsymbol{e}}_{i,t}^{(s)}\big), \qquad \boldsymbol{n}_{i,t}^{(s)} = \mathrm{GRU}_b^{(s)}\big(\tilde{\boldsymbol{e}}_{i,T_i}^{(s)}, \ldots, \tilde{\boldsymbol{e}}_{i,t}^{(s)}\big). \tag{14}$$

Following RETAIN (Choi et al., 2016), the first recurrent stream produces scalar visit-level attention:

$$a_{i,t}^{(s)} = \left[\underset{t' \in \{1,\ldots,T_i\}}{\mathrm{softmax}} \left(\boldsymbol{q}_a^{(s)\top} \boldsymbol{m}_{i,t'}^{(s)}\right)\right]_t, \tag{15}$$

while the second produces feature-level attention:

$$\boldsymbol{b}_{i,t}^{(s)} = \tanh\left(\boldsymbol{W}_b^{(s)} \boldsymbol{n}_{i,t}^{(s)}\right). \tag{16}$$

The context representation for stream s is

$$\boldsymbol{c}_i^{(s)} = \sum_{t=1}^{T_i} a_{i,t}^{(s)}\, \boldsymbol{b}_{i,t}^{(s)} \odot \tilde{\boldsymbol{e}}_{i,t}^{(s)}. \tag{17}$$

The three stream contexts are concatenated and passed to the task head:

$$\hat{Y}_i = f_\theta\left(\left[\boldsymbol{c}_i^{(d)}; \boldsymbol{c}_i^{(p)}; \boldsymbol{c}_i^{(m)}\right]\right). \tag{18}$$

The head is a linear map $\mathbb{R}^{3d_e} \to \mathbb{R}$ for readmission prediction and $\mathbb{R}^{3d_e} \to \mathbb{R}^K$ for medication recommendation. Dropout of 0.5 is applied to stream inputs and dropout of 0.1 to the fused inputs.

A.1.3 LOW-RANK ADAPTATION

We adapt the LLM using LoRA (Hu et al., 2021). The adapter rank is $r_{\mathrm{LoRA}} = 16$, the scaling factor is 32, and no adapter dropout is used. During Stage 2, adapters are restricted to the final four of the $L = 32$ backbone blocks, with updates driven by the prediction loss through the representation readout. After Stage 2, these adapters are merged into the backbone.

Stage 3 attaches fresh LoRA adapters across all 32 blocks. Their B-matrices are initialized to zero, so the initial policy preserves the merged Stage-2 checkpoint for identical inputs. Stage-3 rollouts generate new reasoning trajectories, whereas Stage 2 optimizes representations of fixed cached trajectories. The rollout representations can therefore differ from those used to train the frozen predictor even before policy updates.

A.1.4 PREDICTION LOSS

Denoting the task-specific prediction by $\hat{Y}_i$ and the corresponding target by Y_i, the prediction loss for patient i is

$$\mathcal{L}_{\mathrm{pred}} = \begin{cases} \mathrm{BCE}\big(\hat{Y}_i, Y_i\big), & \text{readmission prediction,} \\ \displaystyle\sum_{k=1}^{K} \mathrm{BCE}\big(\hat{Y}_{i,k}, Y_{i,k}\big), & \text{medication recommendation,} \end{cases} \tag{19}$$

where K is the number of medication labels and BCE denotes binary cross-entropy.

A.2 PROMPTS AND OUTPUT FORMAT

A.2.1 STAGE-1 TARGET CONSTRUCTION

The Stage-1 reconstruction target $\mathcal{T}_{i,t}$ is generated from the structured codes of visit $V_{i,t}$. Each diagnosis, procedure, and medication code is first mapped to its official name or description. A teacher LLM (GPT-5.4 Mini) then converts these codes and descriptions into a concise natural-language summary of the information represented by the structured visit. The teacher receives only

Stage-1 teacher prompt for constructing $\mathcal{T}_{i,t}$

```
System.
You are given the structured medical codes from one clinical visit together with their
official names/descriptions.

Explain the clinically important information represented by these codes.

Requirements:
- use only information supported by the provided codes/names;
- do not infer facts that are not represented;
- do not use any future outcome information;
- be concise but cover all important code groups;
- preserve clinically meaningful relationships when they are directly supported.

Write flowing prose of at most 5 sentences and at most 150 words.  Cover the diagnoses
first, then the procedures, then the medications, and give each of the three a fair share of
the length.  Group medications by class rather than listing every agent.  Never reproduce a
code identifier, never output a bulleted or numbered list, and never invent a code that is
not listed below.

User.
Codes and names:
⟨diagnosis, procedure, and medication codes with official names⟩

Explanation (prose only, no codes, no lists):
```

Figure 3: Teacher prompt used to construct the Stage-1 reconstruction target $\mathcal{T}_{i,t}$ in Equation 4.

Stage-2/3 reasoning instruction $\mathcal{I}$

```
System.
You are a clinical representation model.  You are given two views of the same visit:  the
structured EHR codes recorded for it, supplied as continuous tokens (diagnoses, procedures,
medications), and the free-text discharge note.

The codes are supplied as continuous vectors, not as text:  there is no list of them in this
prompt, and you should not look for one, ask about one, or comment on its absence.  Assume
they cover the visit's coded diagnoses, procedures, and medications.

Those codes already reach the downstream predictor on their own.  Your job is to reason
about what they cannot express.

Think about clinically important information that is in the NOTE but is NOT recoverable from
the coded diagnoses, procedures, and medications---severity and trajectory, why something
happened, response or failure to respond to treatment, key findings and values, functional
and social context, and disposition or follow-up risk.  Reason in your own words.

Two constraints:
- do not copy or transcribe passages of the note;
- do not predict or reveal the ground-truth outcome label.

User.
Task:
Predictive representation for:  ⟨task⟩

Clinical note:
⟨note S_{i,t}⟩

The structured EHR codes for this same visit are supplied here as continuous vectors rather
than as text:
⟨soft prompt E_{i,t}, P continuous tokens⟩

Reason about what this note adds beyond the coded diagnoses, procedures, and medications
represented by the continuous tokens above, then state your conclusion.  Do not copy the
note.  Do not predict the outcome.
```

Figure 4: The instruction $\mathcal{I}$ used in Stages 2 and 3 (Equation 5).

the codes and their official descriptions; it does not receive the clinical note, prediction label, or any future information. The generated targets are cached and reused throughout training. The exact teacher prompt used to construct $\mathcal{T}_{i,t}$ is shown in Figure 3.

A.2.2 Reasoning instruction

The same instruction is used for both prediction tasks; only the ⟨*task*⟩ slot changes between "hospital readmission" and "medication recommendation". The exact instruction is shown in Figure 4.

A.2.3 DELIBERATION-THEN-SUMMARY FORMAT

A trace $R_{i,t}$ consists of a deliberation segment inside `<think>...</think>` followed by a free-standing summary block. The deliberation is bounded by $B = 512$ tokens. If the model does not close the segment within the budget, `</think>` is inserted and generation proceeds to the summary, which is bounded by 448 tokens.

The format reward evaluates the summary block structurally:

$$r_{\text{fmt}} = \begin{cases} -1, & \text{empty summary block,} \\ -0.5, & \text{summary transcribes the note,} \\ +1, & \text{summary terminates itself within the budget,} \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

A summary is classified as transcription when more than 85% of its distinct content words already occur in the note. The trajectory-level $r_{\text{fmt}}^{(g)}$ in Equation 8 is the average of these visit-level scores.

A.3 POLICY ROLLOUT DETAILS

The Stage-3 policy starts from the Stage-2 checkpoint, with Stage-2 adapters merged into the backbone and fresh Stage-3 adapters initialized to preserve its outputs. The reference policy π_{ref} is evaluated with the Stage-3 adapters disabled; the merged Stage-2 adaptations remain active. Reference log-probabilities are detached, and the downstream predictor remains frozen.

For each patient i, rollout g independently resamples reasoning for all historical visits:

$$R_{i,t}^{(g)} \sim \pi_\psi\left(\cdot \mid \boldsymbol{E}_{i,t}, S_{i,t}, \mathcal{I}\right), \qquad t = 1, \ldots, T_i. \tag{21}$$

Each sampled trace is converted to $\boldsymbol{z}_{i,t}^{(g)}$ using Equation 6. The frozen predictor then evaluates the complete patient sequence and produces $\mathcal{L}_{\text{aug}}^{(g)}$.

Within each rollout group, the structured record, soft prompts, notes, labels, and predictor are fixed; only the sampled reasoning trajectories differ. The implementation uses a linear improvement reward $\gamma(\mathcal{L}_{\text{base}} - \mathcal{L}_{\text{aug}}^{(g)})$, where $\mathcal{L}_{\text{base}}$ is the frozen predictor's loss on cached Stage-2 representations. Equation 8 omits the group-constant term $\gamma\mathcal{L}_{\text{base}}$, which cancels under Equation 10 and leaves the normalized advantages unchanged.

The group-relative normalization follows GRPO (Shao et al., 2024). GRPO constructs relative advantages from multiple generations for the same input rather than fitting a separate value model (Shao et al., 2024). In CERE, the grouping unit is a patient, and each group contains G complete patient-level reasoning trajectories.

The token-level likelihood ratio used in Equation 11 is

$$\rho_\ell^{(g)} = \frac{\pi_\psi\left(o_\ell^{(g)} \mid \boldsymbol{x}_\ell^{(g)}\right)}{\pi_{\text{old}}\left(o_\ell^{(g)} \mid \boldsymbol{x}_\ell^{(g)}\right)}, \tag{22}$$

where $\boldsymbol{x}_\ell^{(g)}$ contains the input context for the corresponding visit and its previously generated tokens. Old log-probabilities and advantages are detached. Each fresh rollout group receives one optimizer step, with no parameter updates during gradient accumulation. Thus, $\rho_\ell^{(g)} = 1$ at gradient evaluation, and PPO clipping is inactive. Equation 11 yields the same policy-surrogate gradient as the clipped objective under this schedule.

The sampled reference-policy penalty is

$$\mathcal{L}_{\text{ref}} = \mathbb{E}_g\left[\frac{1}{|R^{(g)}|}\sum_{\ell=1}^{|R^{(g)}|}\left(\exp(u_\ell^{(g)}) - u_\ell^{(g)} - 1\right)\right],$$

where

$$u_\ell^{(g)} = \log \pi_{\text{ref}}\left(o_\ell^{(g)} \mid \boldsymbol{x}_\ell^{(g)}\right) - \log \pi_\psi\left(o_\ell^{(g)} \mid \boldsymbol{x}_\ell^{(g)}\right).$$

Reference log-probabilities are detached, and sampled tokens are held fixed during differentiation. The penalty is nonnegative and vanishes when current and reference probabilities agree on a sampled

Algorithm 1 Optimization procedure for CERE.

Require: Training data $\mathcal{D}$; pretrained LLM $\mathcal{M}$; code encoder g_ϕ; predictor f_θ; instruction $\mathcal{I}$; cached teacher descriptions $\{\mathcal{T}_{i,t}\}$; group size G; number of policy updates N
Ensure: Optimized parameters (ϕ, ψ, θ)

Stage 1: Structured Context Grounding
Freeze all parameters of $\mathcal{M}$
repeat
  Sample a mini-batch of training visits (i, t) ▷ reshuffled each epoch
  Per anchor (i, t), pick a mismatched in-batch visit (i', t') with $i' \neq i$
  $\boldsymbol{E}_{i,t} \leftarrow g_\phi(V_{i,t})$ via Eq. 3
  Compute $\mathcal{L}_{\text{align}}$ via Eq. 4; update ϕ only
until convergence
Freeze ϕ

Stage 2: Residual Evidence Learning
for each training visit (i, t) **do**
  Generate $R_{i,t}$ greedily with $\mathcal{M}$ (no adapters) via Eq. 5; cache ▷ fixed in Stage 2
end for
(i) Cache $\boldsymbol{z}_{i,t}$ via Eq. 6; train θ alone with $\mathcal{L}_{\text{pred}}$ for a fixed number of epochs
(ii) Attach LoRA adapters ψ to the final four transformer blocks of $\mathcal{M}$
for a fixed number of epochs **do**
  Recompute $\boldsymbol{z}_{i,t}(\psi)$ via Eq. 6; predict with the fused RETAIN predictor f_θ
  Update (θ, ψ) jointly via Eq. 7 ▷ θ initialized from (i)
end for ▷ <embed> embedding, base weights, g_ϕ, traces frozen
Freeze θ ▷ frozen judge for Stage 3
Merge Stage-2 adapters into the backbone; fix π_{ref} at this checkpoint
Cache $\boldsymbol{z}^{\text{ref}}_{i,t}$ for all training visits ▷ pre-refinement representations
Attach fresh LoRA adapters ψ to all blocks with zero B-matrices ▷ $\pi_\psi = \pi_{\text{ref}}$ at init

Stage 3: Predictive-Guided Reasoning Refinement
for $n = 1$ **to** N **do**
  Sample a training patient i (without replacement within an epoch)
  Fix π_{old} at the current pre-update policy (log-probabilities detached)
  for $g = 1$ **to** G **do**
    for $t = 1$ **to** T_i **do**
      Sample $R^{(g)}_{i,t} \sim \pi_\psi(\cdot \mid \boldsymbol{E}_{i,t}, S_{i,t}, \mathcal{I})$; compute $\boldsymbol{z}^{(g)}_{i,t}$ via Eq. 6
    end for
    Evaluate frozen f_θ on $\{\boldsymbol{z}^{(g)}_{i,t}\}_{t=1}^{T_i}$ to obtain $\mathcal{L}^{(g)}_{\text{aug}}$
    Compute $r^{(g)}_{\text{fmt}}$, drift $d^{(g)}$ (Eq. 9), and reward $r^{(g)}$ (Eq. 8)
  end for
  Compute advantages $A^{(g)}$ via Eq. 10 (detached)
  Accumulate gradients of $\mathcal{L}_{\text{GRPO}}$ (Eq. 11) across the rollout group
  Take one optimizer step on ψ only ▷ θ, ϕ, base weights frozen
end for

Final Predictor Refit
Freeze ψ
for each training visit (i, t) **do**
  Regenerate $R_{i,t}$ greedily under π_ψ; recompute $\boldsymbol{z}_{i,t}$ via Eq. 6
end for
Reinitialize θ; train θ from scratch with $\mathcal{L}_{\text{pred}}$ for a fixed number of epochs
return (ϕ, ψ, θ)

token. Rollouts use the decoding settings in Table 4; we do not assume that this sampled penalty is an unbiased estimate of the full-policy KL divergence.

The reference-policy penalty acts on token probabilities, whereas the representation penalty discourages deviations beyond the specified tolerance in the frozen predictor's input space. Neither term guarantees the absence of reward exploitation.

Table 4: Hyperparameters and optimization settings of CERE. Values are shared across both tasks. Batch sizes are counted in visits for the alignment objective and in patients for prediction and policy optimization.

Hyperparameter	**Symbol**	**Value**
LLM hidden dimension	D	2560
LLM layers	L	32
Soft-prompt tokens	P	6 (2 per code type)
Optimizer		AdamW
Weight decay		10^{-2}
Gradient-norm clipping		1.0
Structured context grounding		
Margin weight	λ	1.0
Epochs		4
Batch size		8 visits $\times$ 3 replicas
Learning rate (ϕ)		10^{-4}
Learning-rate schedule		cosine annealing
Conditional residual representation learning		
Deliberation budget	B	512 tokens
Summary length		448 tokens
Decoding		greedy
Repetition penalty		1.1
Visit-embedding width	d_e	256
Projected reasoning width	d_z	256
LoRA rank	r_{LoRA}	16
Predictor epochs		11
Predictor batch size		32 patients
Predictor learning rate (θ)		10^{-4}
Encoder fine-tuning epochs		3
Encoder fine-tuning batch size		32 patients (accum.)
Adapter learning rate (ψ)		10^{-5}
Learning-rate schedule		constant
Predictive policy optimization		
Group size	G	8
Prediction-loss weight	γ	0.5
Format-reward weight	α	0.02
Drift-guard weight	δ	0.5
Drift tolerance	τ_d	1.1
Reference-penalty weight	β	0.02
Advantage floor	$\sigma_{\min}$	0.05
Policy learning rate (ψ)		10^{-4}
Learning-rate schedule		constant
Policy updates		200 steps
Rollout temperature		0.9
Top-p		0.95
Repetition penalty		1.1

During Stage 3, only the adapter parameters ψ are updated. The predictor θ, structured encoder ϕ, and LLM base weights remain frozen.

A.4 Training Algorithm

Algorithm 1 summarizes structured context grounding, conditional residual representation learning, policy refinement, and the final predictor refit.

A.5 Implementation Details

Stage 1, structured context grounding, trains g_ϕ on all visits of the training patients (patient-level 80/20 split), data-parallel over three replicas with a per-replica batch of 8 visits. The mismatched visit of Equation 4: For each anchor visit (i, t), the mismatched target in Equation 4 is selected from a visit belonging to a different patient $(i' \neq i)$. The implementation samples one such negative within the batch and resamples it after each epoch shuffle.

Stage 2, conditional residual representation learning, proceeds in two sub-phases that together implement Equation 7. The trace of every training visit is first generated once by greedy decoding and cached. Then (i) the predictor parameters θ (the per-stream attention layers, the fusion maps $\{\boldsymbol{W}_z, \boldsymbol{W}_f^{(s)}\}$, and the head) are trained alone for 11 epochs on the cached readouts, with the LLM, the code encoder g_ϕ, and the traces frozen; (ii) starting from that predictor, the adapters ψ are trained jointly with θ for 3 further epochs (adapters at 10^{-5}, predictor at 10^{-4}). The readout of Equation 6 is recomputed under the adapters at every step, and gradients accumulate from micro-batches of one patient to an effective batch of 32. The `<embed>` input embedding, LLM base weights, code encoder g_ϕ, and cached traces remain frozen throughout both sub-phases.

In Stage 3, patient groups are drawn from the shuffled training split without replacement within an epoch, with one patient group per optimizer step for 200 steps. Each group contains $G = 8$ complete patient-level rollouts. Gradients accumulate across micro-batches before a single optimizer step. Old log-probabilities are evaluated under the pre-update parameters and detached, so the likelihood ratio equals one at gradient evaluation and PPO clipping is inactive. The reference-policy penalty uses the $\exp(u) - u - 1$ formulation in Appendix A.3.

For the final refit, every training visit's trace is regenerated greedily under the refined policy. The predictor θ is retrained from scratch for 11 epochs with a batch size of 32 and a learning rate of 10^{-4}, while ψ remains frozen.

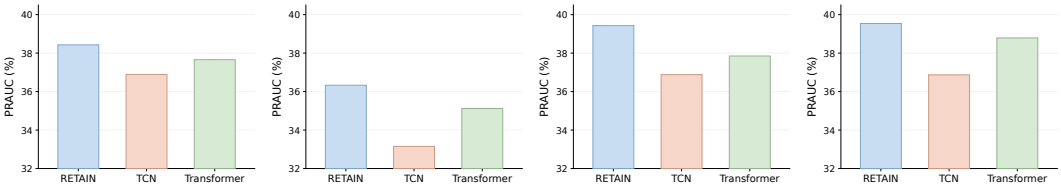
For the variant without RL refinement, we retain the frozen Stage-2 representation model, regenerate representations under the Stage-2 checkpoint using the same decoding procedure, and retrain the downstream predictor from scratch with the same settings used for CERE. This comparison isolates the effect of predictive policy refinement.

A.6 Hyperparameters

Table 4 summarizes the optimization, decoding, and hyperparameter settings used across all stages. Hyperparameters were selected on held-out training data and fixed before evaluation.

A.7 Baselines

We compare CERE against a broad set of deep learning and LLM-based methods for clinical prediction. The deep learning baselines include **RETAIN** Choi et al. (2016), which models longitudinal EHRs using reverse-time recurrent attention to identify influential visits and clinical variables; **DeepR** Nguyen et al. (2017), which represents patient histories as sequences of coded clinical events and applies convolutional neural networks for prediction; **GRAM** Choi et al. (2017), which incorporates hierarchical medical ontologies into EHR representation learning by propagating information from medical concepts to their ancestors; **Transformer** Vaswani et al. (2017), which applies self-attention to capture dependencies across longitudinal clinical events; **AdaCare** Ma et al. (2020), which captures multi-scale temporal variations in clinical features and adaptively emphasizes patient-specific risk factors; **StageNet** Gao et al. (2020), which introduces stage-aware recurrent and convolutional modeling to capture disease progression over time; **GRASP** Zhang et al. (2021), which improves EHR prediction by exploiting representations of clinically similar patients; **ADORE** Cheong et al. (2023), which adapts ontology-derived medical concept representations to EHR data through attention-based integration; and **ARCI** H Moghaddam et al. (2024), which disentangles coexisting temporal medical intents across sequential patient visits. We additionally compare with three recent LLM-based EHR methods: **GraphCare** Jiang et al. (2024), which combines LLM-derived knowledge with external biomedical knowledge graphs to construct patient-specific graphs for GNN-based prediction; **LINKO** Nayebi Kerdabadi et al. (2025), which uses LLM-initialized medical concept representations together with dual-axis propagation across



(a) III-Readmission **(b)** III-Medication **(c)** IV-Readmission **(d)** IV-Medication

Figure 5: Effect of the prediction architecture on CERE. PRAUC is reported using RETAIN, TCN, and Transformer predictors across (a) MIMIC-III readmission, (b) MIMIC-III medication recommendation, (c) MIMIC-IV readmission, and (d) MIMIC-IV medication recommendation.

Visit: 83-year-old admitted with hypotension and pneumonia on chronic heart failure (EF 20%), discharged to an extended-care facility.

Before

Summary: "The note details a critical period where the patient experienced profound hemodynamic instability (hypotension to 70s) and metabolic derangement (lactate elevation), which resolved rapidly with fluid resuscitation… the narrative explains their temporary cessation due to renal function fluctuations and subsequent restart based on blood pressure trends, highlighting the dynamic management required… Finally, the disposition to an Extended Care Facility suggests a high-risk transition…" *(risk factor arrives last, unranked)*

After (GRPO)

Deliberation: "Key factors influencing readmission often include: 1. Severity of illness… 3. Functional status/Social support… 4. Discharge disposition…"

Summary: "**Discharge Disposition:** The patient was discharged to an Extended Care Facility (Rehab). This is a strong predictor of subsequent hospitalization… Codes might capture the diagnosis of heart failure or pneumonia, but they do not capture the severity of the decompensation (e.g., shock/hypotension) which often correlates with higher readmission rates… **Conclusion:** The most critical uncodable factor for predicting readmission here is the discharge destination to an Extended Care Facility, combined with the severity of the initial presentation (hypotension/shock)…"

Figure 6: **Task reframing.** The same clinical content is reorganized from a narrative of the hospital course into an argument about readmission risk: disposition and severity are promoted from closing remarks to ranked, concluded factors. Colors as in Fig. 2.

multiple ontology graphs; and **RePrompT** H Moghaddam et al. (2026), which integrates structured longitudinal EHR encoders with doctoral notes by injecting cohort-level clinical representations using trainable soft prompts and note summarizations as hard prompts. We additionally evaluate four frozen note encoders whose embeddings are fused with structured code embeddings in a RETAIN predictor: **BERT** (Devlin et al., 2019), a bidirectional Transformer pretrained to capture contextual text representations; **ModernBERT** (Warner et al., 2025), an encoder with architectural improvements for efficient long-context processing; **BioClinical-MBERT** (Sounack et al., 2025), a ModernBERT variant further pretrained on biomedical and clinical text; and **Qwen3-Embedding-4B** (Zhang et al., 2025), a four-billion-parameter model optimized for text embedding tasks. All text encoders remain frozen during downstream training. These baselines assess the predictive value of pretrained note representations without explicit code-conditioned reasoning.

A.8 Effect of the Prediction Architecture

Figure 5 compares RETAIN, Transformer, and TCN as predictors within CERE. RETAIN performs best in every setting, followed by Transformer and TCN. Its attention over visits and features (Choi et al., 2016) may be well suited to patient histories in which only some visits and representation dimensions are relevant to the target. This offers a plausible explanation for its advantage when aggregating structured and residual evidence. The Transformer comes closer to RETAIN for medication recommendation on MIMIC-IV than on MIMIC-III, although the ranking remains unchanged. These results favor RETAIN for the evaluated configuration. A comparison against the correspond-

ing code-only baseline for each predictor would be necessary to determine whether the benefit of residual representations persists across architectures.

A.9 Additional Case Studies

Figure 6 presents an additional test-fold example, with greedy decoding of the same visit before and after GRPO. It illustrates task reframing for a patient with pneumonia and chronic heart failure. Before optimization, the rationale recounts the hospital course and mentions discharge to an extended-care facility only at the end. After optimization, it places discharge disposition and initial hemodynamic instability at the center of its readmission analysis. This example illustrates a change in evidence prioritization. It does not independently establish improved predictive accuracy or the absence of the selected information from the structured inputs.