Leaps Beyond the Seen: Reinforced Reasoning Augmented Generation for Clinical Notes

Lo Pang-Yun Ting[♠†], **Chengshuai Zhao**[♥†], **Yu-Hua Zeng**[♠], **Yuan Jee Lim**[♠],
Kun-Ta Chuang[♠], **Huan Liu**[♥]
[♠]Dept. of Computer Science and Information Engineering, National Cheng Kung University
[♥]School of Computing and Augmented Intelligence, Arizona State University
{lpyting, yhzeng, yjlim}@netdb.csie.ncku.edu.tw,
ktchuang@mail.ncku.edu.tw, {czhao93, huanliu}@asu.edu

Abstract

Clinical note generation aims to produce free-text summaries of a patient's condition and diagnostic process, with discharge instructions being a representative long-form example. While recent LLM-based methods pre-trained on general clinical corpora show promise in clinical text generation, they fall short in producing long-form notes from limited patient information. In this paper, we propose ***ReinRAG***, a reinforced reasoning augmented generation (RAG) for long-form discharge instructions based on pre-admission information. *ReinRAG* retrieves reasoning paths from a medical knowledge graph to provide explicit semantic guidance to the LLM. To bridge the information gap, we propose group-based retriever optimization (GRO) which improves retrieval quality with group-normalized rewards, encouraging reasoning leaps for deeper inference by the LLM. Comprehensive experiments on the real-world dataset show that *ReinRAG* outperforms baselines in both clinical efficacy and natural language generation metrics. Further analysis reveals that *ReinRAG* fills semantic gaps in sparse input scenarios, and retrieved reasoning paths help LLMs avoid clinical misinterpretation by focusing on key evidence and following coherent reasoning.

1 Introduction

Clinical note generation improves communications and decision-making among physicians and patients, while also reducing the time burden of manually writing reports [2, 34]. This has motivated research into using large language models (LLMs) for automatic clinical note and report generation [1, 13, 23]. Nevertheless, most works focus on generating short summaries that address specific elements, such as diagnoses or treatments, instead of producing extensive and in-depth outputs.

Patient discharge instruction summarizes a wide range of information, including diagnoses, medications, and the patient's condition during hospitalization [16, 37, 36, 16], while also providing guidance for post-discharge care [15, 10]. Automatically generating discharge instructions can reduce the workload for clinicians. Moreover, generating preliminary discharge instructions could provide clinicians with an early snapshot of likely diagnoses, treatments, and follow-up needs, serving as a useful reference throughout the hospital stay. Despite its significance, *the automatic generation of discharge instructions from pre-admission information* remains largely underexplored and faces following challenges.

[†] Equal contribution.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: The Second Workshop on GenAI for Health: Potential, Trust, and Policy Compliance.

Challenge 1: Open-ended generation without explicit evidence. Generating discharge instruction is inherently an open-ended generation task, where the correct content may not be explicitly present in the data. Most medical LLMs [30, 43] or Retrieval-Augmented Generation (RAG) [19] models [52, 26, 46] are pre-trained on general clinical corpora and are mainly designed for answering questions with explicit evidence or solving closed-ended tasks with predefined answer choices. As a result, they may not be well suited to our scenario.

Challenge 2: Information gap in discharge instructions. There is a significant information gap between patients' pre-admission data and discharge instructions, as the latter typically relies on hospital-stay information. Without proper guidance, LLMs only generate semantically similar content from pre-admission inputs and fail to infer deeper clinical states.

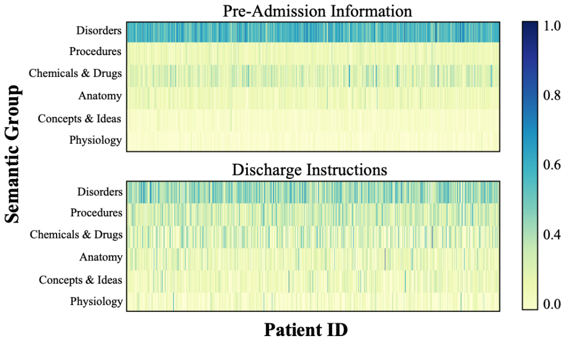


Figure 1: Keyword distribution across UMLS semantic clusters in patients' pre-admission information and discharge instructions. Keywords from pre-admission information are concentrated in the *Disorders* cluster, whereas those in discharge instructions span a broader range of semantic clusters, revealing a substantial information gap.

To analyze information gap in the second challenge, we select 500 de-identified patients' discharge summaries from MIMIC-IV-note [14, 9], which contains data from the Beth Israel Deaconess Medical Center. For each patient, we define pre-admission information as allergies, chief complaints, and the history of present illness (HPI), and compare it with their discharge instructions (an example is shown in Appendix C.1). We then extract keywords from each text and map them to semantic clusters[†] in the UMLS (Unified Medical Language System) [3, 27], which is a comprehensive medical knowledge base structured as a large-scale knowledge graph (KG). Figure 1 presents the distribution of extracted keywords across UMLS semantic clusters, revealing the substantial content difference between the patients' pre-admission information and their discharge instructions. This indicates that, LLMs need to be guided on when to perform fine-grained reasoning [54, 55, 25] to infer more details from known situations (e.g., patient symptoms), and when to perform jump thinking to infer deeper information (e.g., diagnoses or treatments) to bridge the information gap.

These challenges suggest that generating accurate instructions involves two key components: **retrieving external knowledge to provide reasoning direction** that guides accurate long-form generation, and **controlling the granularity of reasoning steps** to help LLMs infer possible downstream clinical details beyond the observed input. In response, we propose the ***ReinRAG*** model (**Rein**forced **R**easoning **A**ugmentation for Clinical Note **G**eneration) for long-form discharge instruction generation based on pre-admission information. **To retrieve useful knowledge and ensure accurate reasoning direction**, we incorporate the UMLS KG to retrieve structured reasoning paths, providing LLMs with explicit semantic guidance in open-ended generation. **To control the LLM's reasoning granularity**, we design a retriever based on reinforcement learning (RL) that learns to select reasoning paths exhibiting reasoning leaps across semantic clusters in the KG. Unlike conventional RAG approaches that rely on single-hop or simple multi-hop retrieval, our method uses RL to optimize the retrieval and guide LLMs on when to retrieve semantically similar concepts or make reasoning leaps to obtain more diverse information. This design helps the LLM advance its reasoning and bridge the information gap when only pre-admission information is available. Furthermore, inspired by Group Relative Policy Optimization (GRPO) [32], we proposed a novel optimization mechanism, named *GRO* (Group-Based Retriever Optimization), which retrieves multiple reasoning paths per patient and assigns group-normalized rewards to discover the most informative semantic paths. Our key contributions are summarized as follows:

- ⋆ **Discharge Instruction Generation with Limited Information.** We target the challenging task of generating long-form discharge instructions using only patients' pre-admission data, going beyond

[†] In this paper, semantic clusters refer to the semantic groups defined in the UMLS semantic network.

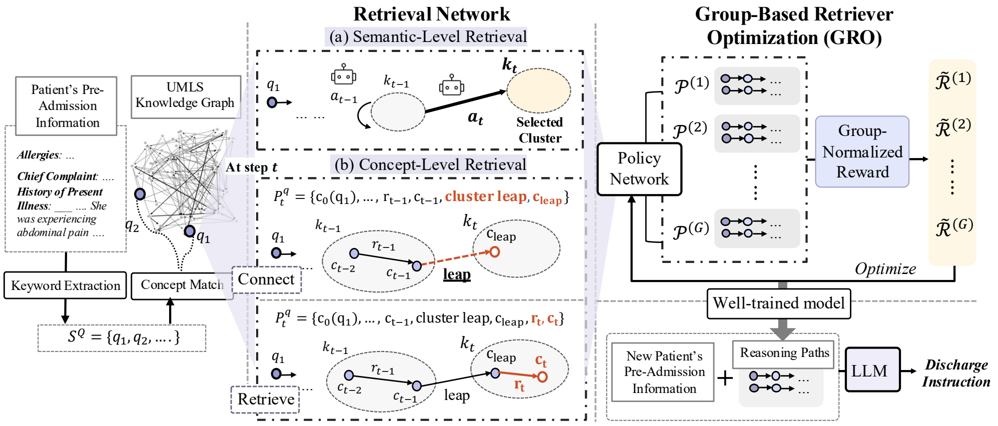

Retrieval Network
(a) Semantic-Level Retrieval
q_1
a_{t-1}
k_{t-1}
k_t
a_t
Selected Cluster
Patient's Pre-Admission Information
UMLS Knowledge Graph
Allergies: ...
Chief Complaint: ....
History of Present Illness: ___ .... She was experiencing abdominal pain ....
q_2
q_1
At step t
(b) Concept-Level Retrieval
$P_t^q = \{c_0(q_1), \dots, r_{t-1}, c_{t-1}, \text{cluster leap}, c_{\text{leap}}\}$
k_{t-1}
r_{t-1}
c_{t-2}
c_{t-1}
c_{leap}
leap
Keyword Extraction
Concept Match
Connect
$S^Q = \{q_1, q_2, \dots\}$
$P_t^q = \{c_0(q_1), \dots, c_{t-1}, \text{cluster leap}, c_{\text{leap}}, r_t, c_t\}$
c_t
r_t
Retrieve
Group-Based Retriever Optimization (GRO)
Policy Network
$\mathcal{P}^{(1)}$
$\mathcal{P}^{(2)}$
$\mathcal{P}^{(G)}$
Group-Normalized Reward
$\tilde{\mathcal{R}}^{(1)}$
$\tilde{\mathcal{R}}^{(2)}$
$\tilde{\mathcal{R}}^{(G)}$
Optimize
Well-trained model
New Patient's Pre-Admission Information
+
Reasoning Paths
LLM
Discharge Instruction

Figure 2: The overview of *ReinRAG*.

conventional short-form generation. This represents a *new and largely unexplored direction with potential clinical value in early decision support*.

- ⋆ **Reinforced Reasoning Augmentation.** We enhance RAG with a novel RL-based retriever that performs *reasoning leaps* across semantic clusters in a medical KG. This guides the LLM to bridge the gap between limited pre-admission inputs and complex discharge instructions, marking a *pioneering application of RL for reasoning-based retrieval in long-form generation*.
- ⋆ **Group-Based Retriever Optimization.** We introduce *GRO*, a novel RL optimization strategy that retrieves multiple reasoning paths per input and leverages group-normalized rewards to effectively guide LLM generation.
- ⋆ **Practical Effectiveness.** Experiments on the real-world MIMIC-IV-note dataset demonstrate that *ReinRAG* consistently outperform baselines in both clinical efficacy and natural language generation, producing more accurate and less irrelevant information.

2 Methodology

The proposed model **Rein**forced **R**easoning **A**ugmentation for Clinical Note **G**eneration, ***ReinRAG***, consists of two main components, as illustrated in Figure 2: (1) *Retrieval Network* (Sec. 2.2), which controls reasoning granularity by performing two-level of retrievals based on RL; and (2) *Group-Based Retriever Optimization* (Sec. 2.3), which optimizes the model based on a group of reasoning paths to guide long-form discharge instruction generation.

2.1 Basic Setup

2.1.1 Notations and Problem Definition

Our goal is to retrieve reasoning paths from a medical KG to guide LLM generation. Formally, a **medical knowledge graph** (UMLS KG [3, 27] used in our paper) is represented as $\mathcal{G} = \{(c, r, c') \mid c, c' \in C, r \in R\}$, where C is the set of medical concepts and R is the set of relations. A triplet (c, r, c') describes the relationship between two concepts, such as ("*dyspnea care*", "*focus of*", "*breathlessness care management*"). Let $\mathcal{G}^k$ denote the set of semantic clusters, where each concept $c \in \mathcal{C}$ belongs to a specific cluster $k \in \mathcal{G}^k$ based on its semantic (e.g., concept "*dyspnea care*" belongs to cluster "*Procedures*"). For **patient information**, let Q be the pre-admission information and S^Q be the set of keywords extracted from Q. Each keyword $q \in S^Q$ can be mapped to a specific concept c in KG $\mathcal{G}$[†].

Each **reasoning path** starts from a keyword $q \in S^Q$ and is denoted as P_t^q at retrieval step t, with $P_0^q = \{q\}$. Therefore, given Q, $\mathcal{G}$, and initial reasoning paths $\{P_0^q\}_{q \in S^Q}$, we aim to retrieve and extend reasoning paths to guide LLM generation.

[†] We describe the extracted terms as "keywords" and KG nodes as "concepts" for clarity.

2.1.2 Retrieval Environment Formulation

Our task is viewed as a Markov Decision Process (MDP), where the retriever decides whether to continue exploring concepts within the current cluster or to leap to another cluster.

State. The state $(s_t^k, s_t^c) \in \mathcal{S}$ represents the current retrieval situation, consisting of the *cluster state* s_t^k and the *concept state* s_t^c, described as follows.

Cluster State s_t^k. The cluster state representation is constructed based on both the currently selected cluster k_t and a scarce cluster k_{scarce}, which is defined as the cluster with the fewest keywords in S^Q. This design encourages the retriever to reason not only within the current cluster but also toward underrepresented semantic. The representation $\mathbf{s}_t^k$ of the cluster state is formulated as:

$$\mathbf{s}_t^k = [\mathbf{k}_t \parallel \mathbf{k}_{\text{scarce}}], \tag{1}$$

where $\mathbf{k}_t \in \mathbb{R}^{2d}$ and $\mathbf{k}_{\text{scarce}} \in \mathbb{R}^{2d}$ denote the hidden state embeddings of k_t and k_{scarce}, respectively. The symbol $\parallel$ represents embedding concatenation.

Concept State s_t^c. The concept state representation is formulated based on all explored concepts, denoted as C_t, as follows:

$$\mathbf{s}_t^c = \mathbf{M} \cdot \text{avg}(\{\mathbf{c} = \text{encoder}(c) | c \in C_t\}), \tag{2}$$

where each concept c is encoded using a pretrained SapBERT encoder [24], which is trained on the UMLS dataset. The matrix $\mathbf{M} \in \mathbb{R}^{d \times d}$ is a learnable projection.

Action. The set of possible actions $A_t \in \mathcal{A}$ at each step t represents "leaps" to another (or the same) clusters in $\mathcal{G}^k$. Formally, an action at step t is defined as $a_t = (k_{t-1} \rightarrow k_t) \in A_t$, indicating the retriever transitions from cluster k_{t-1} to k_t. Each action is represented by the embeddings of the previously visited and the selected clusters, formulated as follows:

$$\mathbf{a}_t = [\mathbf{k}_{t-1} \| \mathbf{k}_t]. \tag{3}$$

After selecting an action, a state transition occurs. The transition function $\delta : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is defined as $\delta((s_t^k, s_t^c), a_t)$, which produces the new state information. Note that at each step, the retriever is allowed to stay in the current cluster or leap to other clusters for the future retrieval. Details of the reward design will be presented in the subsequent sections.

2.2 Retrieval Network

Our retriever aims to retrieve reasoning paths from KG $\mathcal{G}$ by controlling reasoning granularity, which involves deciding when to apply reasoning leaps across semantic clusters (semantic-level) and when to select semantically similar concepts in the current cluster (concept-level), forming the two levels of retrieval process, as shown in Figure 2.

Semantic-Level Retrieval. Following the RL paradigm, our retrieval process is guided by a policy network π_θ, which determines *which semantic cluster to visit next* based on the current state information (s_t^k, s_t^c), as show in Figure 2(a)). To align state and action embeddings so that the policy π_θ can effectively score their semantic compatibility in a shared representation space, we first map the concatenated state representations $[\mathbf{s}_t^k || \mathbf{s}_t^c]$ through a two-layer feedforward network to obtain a hidden representation $\mathbf{z}_t$. Based on $\mathbf{z}_t$, the policy distribution $\mathbf{d}_t$ over possible actions A_t is then computed, reflecting the probability of selecting each action at step t given the current states. Hidden representation $\mathbf{z}_t$ and policy distribution $\mathbf{d}_t$ are defined as:

$$\mathbf{z}_t = \mathbf{W}_2\text{ReLU}(\mathbf{W}_1[\mathbf{s}_t^k || \mathbf{s}_t^c]), \quad \mathbf{d}_t = \pi_\theta(\cdot | s_t^k, s_t^c) = \text{softmax}(\mathbf{A}_t \mathbf{z}_t), \tag{4}$$

where $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{4d \times 4d}$ are the learnable weights, $\mathbf{A}_t \in \mathbb{R}^{|A_t| \times 4d}$ represent the embeddings of next possible actions A_t. The action a_t at step t is then selected as:

$$a_t \sim \text{categorical}(\mathbf{d}_t). \tag{5}$$

Concept-Level Retrieval. Once the next semantic cluster k_t is selected, the retriever proceeds to identify concepts within this cluster to extend the reasoning paths. This step grounds the high-level cluster selection in a concrete concept-to-concept transition within the medical KG. We mainly have two actions for retrieving concepts in the selected cluster k_t, as shown in Figure 2(b).

`Connect`. If the selected cluster k_t differs from k_{t-1}, we first establish a connection between them. Let C_{cand} denote the set of concepts in k_t that appear in previously explored paths $\{P_{t-1}^q\}_{q \in S^Q}$. For each path $P_{t-1}^q = \{c_0(q), ...r_{t-1}, c_{t-1}\}$, we select a connection point $c_{\text{leap}} \in C_{\text{cand}}$ and link the new cluster through c_{leap}. The point is chosen based on the maximum cosine similarity with the path embedding: $c_{\text{leap}} = \arg\max_{c \in C_{cand}}(\text{sim}(\mathbf{c}, \mathbf{P}_{t-1}^q))$, where $\mathbf{c}$ is the embedding of concept c, and $\mathbf{P}_{t-1}^q$ is the average embedding of concepts in path P_{t-1}^q. The path is then updated as $P_t^q = P_{t-1}^q \cup \{\text{“cluster leap”}, c_{\text{leap}}\}$.

`Retrieve`. After establishing the connection, we retrieve new concepts by selecting c_{leap}'s neighbors $N(c_{\text{leap}})$ in cluster k_t. These new concepts provide semantically novel yet coherent information that extends and supports the prior reasoning path, thereby guiding the LLM to perform reasoning leaps and draw deeper inferences. Let $\mathbf{S}^Q$ and $\mathbf{P}_t^q$ denote the average embeddings of keywords in S^Q and concepts in path P_t^q, respectively. The new concept c_t to be added to P_t^q is selected as:

$$c_t = \arg \max_{c' \in N(c_{\text{leap}})} \big[(\mathbf{c}', \mathbf{S}^Q), (\mathbf{c}', \mathbf{P}_t^q)\big]_{\text{sim}}, \tag{6}$$

where $[\cdot, \cdot]_{\text{sim}}$ denotes the average of the cosine similarities between the two pairs of embeddings. $\mathbf{c}'$ is the embedding of candidate concept c'. Therefore, the path is updated as $P_t^q = P_t^q \cup \{r_t, c_t\}$, where r_t denotes the relation connecting c_{leap} and c_t in KG $\mathcal{G}$.

2.3 GRO: Group-Based Retriever Optimization

To ensure that the retrieved paths can enhance LLM generation, a reward is provided when the reasoning paths reach the predefined length. This delayed feedback (episodic reward) allows the model to evaluate the overall quality of complete paths in supporting long-form instruction generation.

Mixture of Rewards. The evaluation of each reasoning path P is based on two criteria: ❶ it contains concepts that appear in the ground-truth instruction, directly contributing to accurate LLM outputs; and ❷ it includes semantically related concepts to guide the LLM toward generating relevant content.

To ensure these objectives are reflected in the episodic rewards, we adopt the following design. First, inspired by recent formulations of verifiable rewards [18, 11], we introduce a binary reward that assigns a value of 1 if the path contains any ground-truth concepts. Second, we incorporate a soft reward based on the embedding similarity between the concepts explored in P and the ground-truth concepts $\hat{C}$. Thus, the reward for reasoning path P is formulated as:

$$R_P = \sum_{c \in P} \mathbb{I}\{c \in \hat{C}\} + \lambda \cdot \text{sim}(\mathbf{P}, \hat{\mathbf{C}}), \tag{7}$$

where λ is a weighting factor. $\mathbf{P}$ and $\hat{\mathbf{C}}$ denote the average embeddings of concepts in P and ground-truth set $\hat{C}$, respectively. $\text{sim}(\cdot, \cdot)$ represents the cosine similarity. $\mathbb{I}\{\cdot\}$ is the indicator function, which returns 1 if c belongs to $\hat{C}$, and 0 otherwise.

Group-Based Optimization. After each episode, the policy network is updated based on the rewards. Inspired by GRPO [32], we adopt its idea of using multiple rollouts per input to estimate the group-normalized reward. Therefore, we propose the *GRO* mechanism (Group-Based Retriever Optimization) to further improve the quality of retrieved paths under sparse episodic rewards. This also stabilizes learning by better attributing credit across entire paths.

Specifically, we perform a fixed number G of retrieval processes for each patient. Let $\mathcal{P}^{(i)}$ denote the path set retrieved in the i^{th} process. After G retrievals, we obtain a reward set $\mathbf{R} = \{\mathcal{R}^{(1)}, ..., \mathcal{R}^{(G)}\}$, where $\mathcal{R}^{(i)} = \sum_{P \in \mathcal{P}^{(i)}} R_P$. The group-normalized reward for each retrieval process is defined as:

$$\tilde{\mathcal{R}}^{(i)} = \frac{\mathcal{R}^{(i)} - \mu^R}{\sigma^R + \epsilon}, \tag{8}$$

where μ^R and σ^R denote the mean and standard deviation of $\mathbf{R}$, respectively, and ϵ is a small constant for numerical stability.

The optimization aims to maximize the expected cumulative return. We revise the REINFORCE algorithm [41] by using discounted cumulative returns based on normalized rewards:

$$J(\theta) = \mathop{\mathbb{E}}_{\{\mathcal{P}^{(i)}\}_{i=1}^{G} \sim \pi_\theta} \left[\frac{1}{G} \sum_{i=1}^{G} \sum_{t=0}^{T-1} \gamma^{(T-t)} \cdot \tilde{\mathcal{R}}^{(i)} \right], \tag{9}$$

where T is maximum path length and $\gamma \in [0,1]$ is the discount factor. To encourage exploration, the entropy term [42] is added: $\beta \mathcal{H}\left(\pi_\theta(\cdot | s_t^{(i)})\right)$, where the state is $s_t^{(i)} = (s_t^{k(i)}, s_t^{c(i)})$. $\mathcal{H}$ denotes policy entropy. $\beta \geq 0$ controls the exploration strength and is decayed during training. Let $\tilde{\mathcal{R}}_t^{(i)} = \gamma^{(T-t)} \cdot \tilde{\mathcal{R}}^{(i)}$ and $\mathcal{H}_t^{(i)}$ short for $\mathcal{H}\left(\pi_\theta(\cdot | s_t^{(i)})\right)$. The policy network π_θ is updated via the gradient of the objective:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\{\mathcal{P}^{(i)}\}_{i=1}^{G} \sim \pi_\theta} \left[\frac{1}{G} \sum_{i=1}^{G} \sum_{t=0}^{T-1} \left(\nabla_\theta \log \pi_\theta(a_t^{(i)} | s_t^{(i)}) \tilde{\mathcal{R}}_t^{(i)} + \beta \nabla_\theta \mathcal{H}_t^{(i)} \right) \right] \tag{10}$$

Finally, given a well-trained retriever with policy $\hat{\pi}_\theta$, KG $\mathcal{G}$, a new patient's pre-admission information Q', and extracted keywords $S^{Q'}$, the reasoning paths $\{P^q\}_{q \in S^{Q'}} \sim \hat{\pi}_\theta$ are retrieved from $\mathcal{G}$. The LLM $\mathcal{M}$ then generates the ideal discharge instruction $\hat{\mathcal{I}}$ using our *ReinRAG* model as follows:

$$\hat{\mathcal{I}} = \text{ReinRAG}(Q'; \mathcal{M}, \hat{\pi}_\theta, \mathcal{G}) = \arg\max_{\mathcal{I}} \mathbb{P}_{\mathcal{M}}\left(\mathcal{I} \mid Q', \{P^q\}_{q \in S^{Q'}} \sim \hat{\pi}_\theta\right). \tag{11}$$

3 Experiments

3.1 Experimental Setup

Dataset and Preprocessing. We conduct experiments on a subset of MIMIC-IV-note [14, 9], which contains 331,794 de-identified discharge summaries from 145,915 patients at the Beth Israel Deaconess Medical Center. We select 4,000 summaries, where 3,000 for training and 1,000 for testing. From each summary, we extract pre-admission information, including allergies, chief complaint, and history of present illness (HPI), which serves as both the model input and the prompt content for the LLMs. For the medical KG, we adopt the UMLS [3, 27], a large-scale resource developed by the National Library of Medicine and structured as a KG with concepts, semantic relations, and semantic clusters (semantic groups). Following [8], we focus on SNOMED CT (Systematized Nomenclature of Medicine–Clinical Terms) concepts and use 107 diagnostic-related relations.

Keyword Extraction and Concept Matching. We use QuickUMLS [35] to extract keywords from patient information and map to UMLS concepts (focus on SNOMED CT). The best-matched concept for each keyword is selected. Neo4j is utilized to retrieve reasoning paths from the UMLS KG.

Baselines. We compare with following baselines:

- **Vanilla LLMs** include LLaMA-3.1-8B-Instruct [6], Qwen2.5-7B-Instruct [48], Qwen-UMLS-7B-Instruct [29], Mistral-7B-Instruct-v0.3 [12], using pre-admission data as prompt for generation.
- **Medical-Domain LLMs** include LLMs pre-trained or instruction-tuned on biomedical literature, clinical notes, or medical QA corpora, including ChatDoctor-7B [21], Med-Alpaca-7B [33], Meditron-7B [4], Biomistral-7B [17], PMC-LLaMA-13B [43], and MMed-Llama-3-8B [30].
- **Retrieved-Based Methods** consider two one-hop neighbor retrieval baselines. Both identify KG concepts structurally connected to keywords extracted from the pre-admission information: one randomly selects one-hop neighbors, denoted as "**Random1hop**", and the other selects those most semantically similar to the full pre-admission input, denoted as "**Sim1hop**". Both baselines retrieve from the KG without performing reasoning leaps or structuring the retrieved information into paths. We also compare with **DR.KNOWS** [8], which performs path-based retrieval on the KG. The retrieved concepts and original input are used to prompt LLMs for generation.

Table 1: CE and NLG valuations (%) of different models. "J" denotes Jaccard similarity, and "HL" represents Hamming loss. "RG", 'BL", "MTR" and "SBERT" denote ROUGE, BLEU, METEOR and Sentence-BERT, respectively. The best and second-best results are highlight in **bold** and underline. The "Δ" column shows the performance difference from the Vanilla LLaMA-3.1-8B-Instruct.

Model ↓	**CE Metrics** (Concept-Level)										**NLG Metrics**									
Metric →	P(↑)	Δ	R(↑)	Δ	F1(↑)	Δ	J(↑)	Δ	HL(↓)	Δ	RG-L(↑)	Δ	BL-2(↑)	Δ	F1_{BERT}(↑)	Δ	MTR(↑)	Δ	SBERT(↑)	Δ
Vanilla LLMs																				
LLaMA-3.1-8B	98.00	-	28.50	-	42.80	-	7.04	-	71.50	-	11.04	-	6.12	-	81.32	-	22.75	-	46.18	-
Qwen2.5-7B	99.20	*(+1.2)*	34.74	*(+6.2)*	50.14	*(+7.3)*	**7.41**	*(+0.3)*	65.26	*(-6.2)*	10.81	*(-0.2)*	6.13	*(+0.0)*	81.34	*(+0.0)*	24.02	*(+1.3)*	47.74	*(+1.6)*
Qwen-UMLS-7B	91.20	*(-6.8)*	18.20	*(-10.3)*	28.69	*(-14.1)*	5.23	*(-1.8)*	81.80	*(+10.3)*	8.36	*(-2.7)*	4.08	*(-2.0)*	79.63	*(-1.7)*	16.16	*(-6.5)*	39.27	*(-6.9)*
Mistral-7B-v0.3	**99.60**	*(+1.6)*	34.24	*(+5.7)*	49.56	*(+6.7)*	7.04	*(+0.0)*	65.76	*(-5.7)*	10.33	*(-0.7)*	5.38	*(-0.7)*	81.18	*(-0.1)*	23.30	*(+0.6)*	43.83	*(-2.4)*
Medical-Domain LLMs																				
ChatDoctor-7B	76.00	*(-22.0)*	11.32	*(-17.2)*	18.86	*(-23.9)*	4.91	*(-2.1)*	88.67	*(+17.2)*	9.53	*(-1.5)*	**6.97**	*(+0.9)*	81.17	*(-0.2)*	13.49	*(-9.3)*	30.93	*(-15.3)*
Med-Alpaca-7B	85.80	*(-12.2)*	16.27	*(-12.2)*	26.09	*(-16.7)*	5.43	*(-1.6)*	83.72	*(+12.2)*	9.64	*(-1.4)*	5.64	*(-0.5)*	80.56	*(-0.8)*	15.75	*(-7.0)*	36.00	*(-10.2)*
Meditron-7B	91.50	*(-6.5)*	14.87	*(-13.6)*	24.58	*(-18.2)*	2.45	*(-4.6)*	85.12	*(+13.6)*	5.66	*(-5.4)*	2.09	*(-4.0)*	77.54	*(-3.8)*	15.21	*(-7.5)*	15.96	*(-30.2)*
Biomistral-7B	53.10	*(-44.9)*	5.30	*(-23.2)*	9.09	*(-33.7)*	2.66	*(-4.4)*	94.69	*(+23.2)*	6.80	*(-4.2)*	2.25	*(-3.9)*	78.15	*(-3.2)*	7.34	*(-15.4)*	20.81	*(-25.4)*
PMC-LLaMA-13B	26.60	*(-71.4)*	3.20	*(-25.3)*	5.37	*(-37.4)*	1.42	*(-5.6)*	96.79	*(+25.3)*	3.54	*(-7.5)*	0.90	*(-5.2)*	67.22	*(-14.1)*	3.91	*(-18.8)*	13.24	*(-32.9)*
MMed-Llama-3-8B	72.90	*(-25.1)*	10.97	*(-17.5)*	17.98	*(-24.8)*	1.93	*(-5.1)*	89.03	*(+17.5)*	3.51	*(-7.5)*	1.41	*(-4.7)*	74.01	*(-7.3)*	10.14	*(-12.6)*	15.25	*(-30.9)*
Retrieval-Based Methods																				
Random1hop																				
+ LLaMA-3.1-8B	98.40	*(+0.4)*	31.82	*(+3.3)*	46.63	*(+3.8)*	7.05	*(+0.0)*	68.18	*(-3.3)*	10.68	*(-0.4)*	5.64	*(-0.5)*	81.49	*(+0.2)*	22.59	*(-0.2)*	48.51	*(+2.3)*
+ Qwen2.5-7B	98.90	*(+0.9)*	34.32	*(+5.8)*	49.73	*(+6.9)*	7.04	*(+0.0)*	65.68	*(-5.8)*	10.50	*(-0.5)*	5.63	*(-0.5)*	81.31	*(-0.0)*	23.50	*(+0.8)*	48.38	*(+2.2)*
+ Qwen-UMLS-7B	86.00	*(-12.0)*	15.99	*(-12.5)*	25.60	*(-17.2)*	4.13	*(-2.9)*	84.01	*(+12.5)*	7.72	*(-3.3)*	3.32	*(-2.8)*	78.93	*(-2.4)*	14.64	*(-8.1)*	35.75	*(-10.4)*
+ Mistral-7B-v0.3	99.10	*(+1.1)*	32.56	*(+4.1)*	47.72	*(+4.9)*	6.76	*(-0.2)*	67.44	*(-4.1)*	10.21	*(-0.8)*	5.12	*(-1.0)*	81.02	*(-0.3)*	22.86	*(+0.1)*	43.83	*(-2.4)*
Sim1hop																				
+ LLaMA-3.1-8B	98.60	*(+0.6)*	30.36	*(+1.9)*	45.03	*(+2.2)*	6.89	*(-0.2)*	69.64	*(-1.9)*	10.71	*(-0.3)*	5.62	*(-0.5)*	81.50	*(+0.2)*	22.50	*(-0.3)*	47.89	*(+1.7)*
+ Qwen2.5-7B	99.30	*(+1.3)*	34.63	*(+6.1)*	50.01	*(+7.2)*	7.13	*(+0.1)*	65.37	*(-6.1)*	10.52	*(-0.5)*	5.67	*(-0.5)*	81.29	*(-0.0)*	23.57	*(+0.8)*	48.31	*(+2.1)*
+ Qwen-UMLS-7B	87.60	*(-10.4)*	16.38	*(-12.1)*	26.20	*(-16.6)*	4.26	*(-2.8)*	83.62	*(+12.1)*	7.81	*(-3.2)*	3.29	*(-2.8)*	78.89	*(-2.4)*	14.76	*(-8.0)*	36.21	*(-10.0)*
+ Mistral-7B-v0.3	99.30	*(+1.3)*	33.31	*(+4.8)*	48.54	*(+5.7)*	6.66	*(-0.4)*	66.69	*(-4.8)*	10.04	*(-1.0)*	4.99	*(-1.1)*	80.96	*(-0.4)*	22.80	*(+0.0)*	43.85	*(-2.3)*
DR.KNOWS																				
+ Flan-T5-Large	54.00	*(-44.0)*	5.13	*(-23.4)*	8.88	*(-33.9)*	2.60	*(-4.4)*	94.87	*(+23.4)*	4.65	*(-6.4)*	3.26	*(-2.9)*	77.43	*(-3.9)*	5.78	*(-17.0)*	23.52	*(-22.7)*
+ LLaMA-3.1-8B	98.10	*(+0.1)*	23.44	*(-5.1)*	36.55	*(-6.2)*	3.44	*(-3.6)*	76.56	*(+5.0)*	4.78	*(-6.3)*	1.85	*(-4.3)*	78.61	*(-2.7)*	14.42	*(-8.3)*	38.54	*(-7.6)*
+ Mistral-7B-v0.3	94.50	*(-3.5)*	17.55	*(-11.0)*	28.61	*(-14.2)*	4.91	*(-2.1)*	82.45	*(+11.0)*	8.93	*(-2.1)*	4.36	*(-1.8)*	80.74	*(-0.6)*	17.21	*(-5.5)*	43.71	*(-2.4)*
Our Model																				
ReinRAG (**ours**)																				
+ Mistral-7B-v0.3	99.20	*(+1.2)*	**40.73**	*(+12.2)*	**56.01**	*(+13.2)*	6.42	*(-0.6)*	**59.27**	*(-12.2)*	**12.07**	*(+1.0)*	5.58	*(-0.5)*	**82.22**	*(+0.9)*	**24.07**	*(+1.3)*	**55.24**	*(+9.1)*

Evaluation Metrics. Models are evaluated with two types of metrics to compare generated and ground-truth discharge instructions:

- **Clinical Efficacy (CE)**: We assess the correctness of the generated instructions by matching keyword (N-gram level) and SNOMED CT concepts (concept level) with concepts from ground-truth instructions, using precision, recall, F1 score, Hamming loss, and Jaccard similarity. These metrics evaluate the correctness of medically relevant word generation.
- **Natural Language Generation (NLG)**: We report ROUGE-1/2/L [22], BLEU-1/2 [28], METEOR [5], BERTScore (F1), and Sentence-BERT [31] similarity scores to measure the fluency and semantic consistency of the generation.

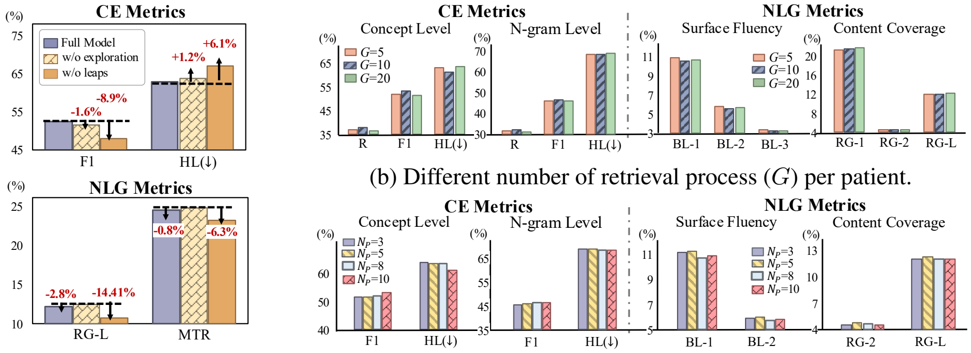
3.2 Comparison Performance

Table 1 reports the CE and NLG performance of all models. Key findings are summarized below:

Clinical Accuracy and Noise Sensitivity. For CE metrics, *ReinRAG* achieves comparable precision, while outperforming vanilla LLaMA and the best baseline (Qwen2.5) by at least 12% and 6%, respectively, in both recall and F1 score. Interestingly, vanilla LLMs sometimes outperform retrieval-based baselines. This suggests that simply retrieving information directly related to pre-admission data can sometimes degrade LLM performance. Medical-domain LLMs, which are pre-trained on clinical corpora for short-form tasks, also fail to improve performance. In contrast, *ReinRAG* achieves the highest F1 score while also reducing Hamming loss by at least 12% and 5% compared to vanilla LLaMA and the best baselines (Qwen2.5), respectively. This indicates that incorporating RL into retrieval can effectively guide LLMs toward accurate long-form generation rather than hindering it.

Semantic Consistency. For NLG quality, *ReinRAG* achieves the highest scores on most metrics. This indicates that our generation preserves the core meaning of ground-truth instructions with less irrelevant descriptions. Although it obtains a lower BLEU-2 score, the highest ROUGE-L, BERTScore (F1_{BERT}), METEOR and Sentence-BERT similarity scores confirm that *ReinRAG* produces outputs that remain semantically similar to ground truths at the paragraph level. This suggests that our generation better captures longer-range overlaps and adheres more closely to ground truths.

Effectiveness of Controlling Reasoning Granularity. Similar to our method, DR.KNOWS [8] also retrieves paths from the KG to prompt LLMs. However, its retrieval is limited to concepts directly connected to the prompt content. This restricts its ability to reason across distant semantic information. As a result, it underperforms *ReinRAG* across all metrics. This demonstrates that *ReinRAG*'s adaptive



(a) Ablation study.

(b) Different number of retrieval process (G) per patient.

(c) Different number of paths (N_P) prompted to the LLM.

Figure 3: In-Depth Analysis of *ReinRAG* with Mistral-7B-Instruct-v0.3

control of reasoning granularity, which allows reasoning leaps, can form more effective paths to better guide LLM generation.

3.3 In-depth Analysis

Ablation Study. We conduct an ablation study by removing (i) the exploration ability of *ReinRAG* (the entropy term in Eq. 10) and (ii) reasoning leaps during retrieval, referred to "w/o exploration" and "w/o leaps", respectively. Results in Figure 3a indicate that removing reasoning leaps significantly degrades both CE and NLG performance. Removing exploration ability slightly improves ROUGE-L and METEOR but leads to lower F1 score and higher Hamming loss compared to the full *ReinRAG*. These results suggest that allowing reasoning leaps effectively guides the LLM toward broader reasoning granularity, helping it generate more accurate information. Meanwhile, the exploration ability of *ReinRAG*, despite slightly sacrificing the semantic consistency with ground truths, improves the LLM to generate more accurate concepts. Proper tuning the exploration strength can further balance and enhance the performance, demonstrating the effectiveness of the *ReinRAG* design.

Parameter Sensitivity Analysis. To evaluate the impact of the number of retrieval processes (G in Eq. 9) and the number of paths prompted to the LLM (denoted as N_P), we vary these parameters to examine the performance. In Figure 3b, setting G to 10 achieves better CE performance. Increasing G to 20 slightly improves ROUGE scores but decreases F1, suggesting that excessive retrievals may introduce noise and harm medical concept correctness, despite slightly improve overall content coverage. Figure 3c indicates that a larger number of prompted paths generally improve CE metrics, but too many paths may also reduce semantic consistency in LLM generation. These results highlight the importance of properly setting both the number of retrievals and prompted paths to balance CE and NLG performance.

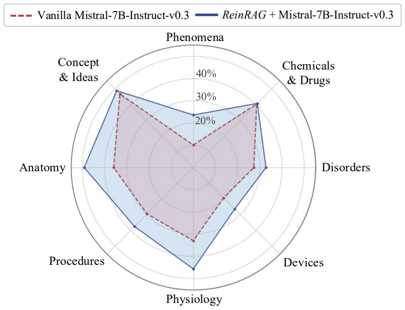


Figure 4: Recall of vanilla Mistral-7B-Instruct-v0.3 and *ReinRAG* model across semantic clusters in the UMLS KG.

3.4 Impact Across Semantic Clusters

To analyze which aspects of generation benefit from *ReinRAG*, we compare the recall of medical concepts generated by *ReinRAG* and vanilla Mistral across eight representative semantic clusters. The results are shown in Figure 4.

Limited Impact in Well-Covered Semantics. In clusters such as *Concepts & Ideas* and *Chemicals & Drugs*, *ReinRAG* shows similar performance to vanilla Mistral. These clusters primarily include non-critical terms (e.g., "Dosing instruction fragment") or explicitly mentioned pre-admission medications. Thus, the vanilla Mistral already achieves high recall in these clusters, suggesting that *ReinRAG* contributes less in these semantic information.

Table 2: Medical professionals' feedback on discharge instructions generated by Vanilla-Mistral-7B-Instruct-v0.3 and our *ReinRAG*.

Model	Strengths	Weakness
Vanilla	*"The care suggestions are detailed and comprehensive, and the instructions are highly related to patients' pre-admission information."*	*"Unrelated medications and diagnostic errors often occurs, such as inappropriate medication or diet suggestions. Most of the diagnostic logic is messy and irrelevant."*
ReinRAG	*"The instructions are more concise and logical, focusing on the core diagnosis and treatments. The number of wrong diagnoses is relatively low."*	*"The instructions are sometimes unclear. There are occasional information errors and omissions in a few cases, though key concepts are mentioned."*

Improved Recall in Information-Sparse Clusters. *ReinRAG* significantly improves recall in clusters like *Anatomy*, *Procedures*, *Physiology*, and *Phenomena*, which include concepts related to body parts, diagnoses, treatments, organ functions, and physiological phenomena. These types of information are typically gathered during a patient's hospital stay and are often underrepresented or implicit in the pre-admission data. This demonstrate that *ReinRAG* effectively bridges the information gap by retrieving reasoning paths from the KG based on known clues.

3.5 Human Evaluation

To verify whether *ReinRAG* can assist clinical practice, we invite two medical processionals to conduct a human evaluation. They review 20 instructions generated by Vanilla Mistral and our *ReinRAG*.

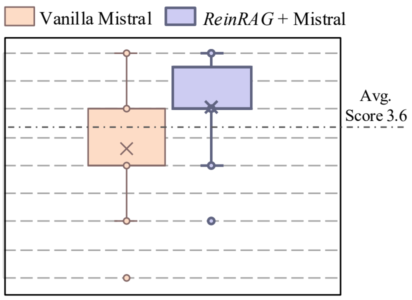


Figure 5: Overall human evaluation performance of Vanilla Mistral-7B-Instruct-v0.3 and *ReinRAG*. Scores range from 1 to 5, with higher scores indicating better performance.

In Table 2, we present representative comments from two medical professionals after they review 20 patient cases. The feedback reveals that vanilla Mistral tends to provide more comprehensive discharge information but often generates irrelevant or event incorrect instructions. In contrast, while *ReinRAG*'s generation occasionally lacks detailed descriptions, the outputs are more accurate and logically reasoned. This suggests that vanilla Mistral, without guidance of our reasoning paths, may provide abundant medical information but often in the wrong direction, failing to align with the patients' actual clinical needs. Moreover, inspired by the evaluation designed in [8], two medical professionals assess the generation based on following aspects: (1) reading comprehension, (2) rationale instructions, (3) rare omission of critical information and (4) minimal irrelevant information. The scores range from 1 to 5, representing strongly disagree, disagree, neutral, agree, and strongly agree, respectively. The overall evaluation scores of both methods are shown in Figure 5, where *ReinRAG* outperforms the vanilla Mistral model, with not only higher average scores but also a narrower value range, indicating more consistent evaluations. This human evaluation highlights the potential of *ReinRAG* to assist clinicians as a reference for early clinical decision-making.

4 Conclusion

This paper introduces *ReinRAG*, a novel RL–based retrieval leveraging reasoning paths to guide LLMs in generating discharge instructions using only pre-admission data. By controlling the reasoning granularity through reasoning leaps and utilizing the proposed GRO, *ReinRAG* retrieves high-quality reasoning paths. Experiments on the MIMIC-IV-Note dataset show that *ReinRAG* outperforms baselines in both clinical efficacy and natural language generation. The human evaluation further validates that *ReinRAG* enables LLMs to avoid clinical misinterpretation and generate accurate and coherent instructions.

Acknowledgments

This paper was supported in part by National Science and Technology Council (NSTC), R.O.C., under Contract 113-2221-E-006-203-MY2, 114-2622-8-006-002-TD1 and 114-2634-F-006-001-MBK.

References

[1] A. B. Abacha, W.-w. Yim, Y. Fan, and T. Lin. An empirical study of clinical note generation from doctor-patient encounters. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2291–2302, 2023.

[2] B. G. Arndt, J. W. Beasley, M. D. Watkinson, J. L. Temte, W.-J. Tuan, C. A. Sinsky, and V. J. Gilchrist. Tethered to the ehr: primary care physician workload assessment using ehr event log data and time-motion observations. *The Annals of Family Medicine*, 15(5):419–426, 2017.

[3] O. Bodenreider. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270, 2004.

[4] Z. Chen, A. H. Cano, A. Romanou, A. Bonnet, K. Matoba, F. Salvi, M. Pagliardini, S. Fan, A. Köpf, A. Mohtashami, et al. Meditron-70b: Scaling medical pretraining for large language models. *arXiv preprint arXiv:2311.16079*, 2023.

[5] M. Denkowski and A. Lavie. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the sixth workshop on statistical machine translation*, pages 85–91, 2011.

[6] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

[7] S. Ellershaw, C. Tomlinson, O. E. Burton, T. Frost, J. G. Hanrahan, D. Z. Khan, H. L. Horsfall, M. Little, E. Malgapo, J. Starup-Hansen, et al. Automated generation of hospital discharge summaries using clinical guidelines and large language models. In *AAAI 2024 Spring Symposium on Clinical Foundation Models*, 2024.

[8] Y. Gao, R. Li, E. Croxford, J. Caskey, B. W. Patterson, M. Churpek, T. Miller, D. Dligach, and M. Afshar. Leveraging medical knowledge graphs into large language models for diagnosis prediction: Design and application study. *JMIR AI*, 4:e58670, 2025.

[9] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000.

[10] D. C. Gonçalves-Bradley, N. A. Lannin, L. M. Clemson, I. D. Cameron, and S. Shepperd. Discharge planning from hospital. *Cochrane database of systematic reviews*, (1), 2016.

[11] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[12] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[13] H. Jin, H. Che, Y. Lin, and H. Chen. Promptmrg: Diagnosis-driven prompts for medical report generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 2607–2615, 2024.

[14] A. Johnson, T. Pollard, S. Horng, L. A. Celi, and R. Mark. Mimic-iv-note: Deidentified free-text clinical notes (version 2.2). physionet, 2023.

[15] A. J. Kind and M. A. Smith. Documentation of mandated discharge summary components in transitions from acute to subacute care. 2011.

[16] I. Kononenko. Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in medicine*, 23(1):89–109, 2001.

[17] Y. Labrak, A. Bazoge, E. Morin, P. Gourraud, M. Rouvier, and R. Dufour. Biomistral: A collection of open-source pretrained large language models for medical domains. In *ACL (Findings)*, pages 5848–5864. Association for Computational Linguistics, 2024.

[18] 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.

[19] 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*, 33:9459–9474, 2020.

[20] R. Li, X. Wang, and H. Yu. Llamacare: An instruction fine-tuned large language model for clinical nlp. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10632–10641, 2024.

[21] Y. Li, Z. Li, K. Zhang, R. Dan, S. Jiang, and Y. Zhang. Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge. *Cureus*, 15(6), 2023.

[22] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[23] C. Liu, Y. Tian, W. Chen, Y. Song, and Y. Zhang. Bootstrapping large language models for radiology report generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18635–18643, 2024.

[24] F. Liu, E. Shareghi, Z. Meng, M. Basaldella, and N. Collier. Self-alignment pretraining for biomedical entity representations. In *NAACL-HLT*, pages 4228–4238. Association for Computational Linguistics, 2021.

[25] J. Liu, J. Lin, and Y. Liu. How much can rag help the reasoning of llm? *arXiv preprint arXiv:2410.02338*, 2024.

[26] A. Lozano, S. L. Fleming, C.-C. Chiang, and N. Shah. Clinfo. ai: An open-source retrieval-augmented large language model system for answering medical questions using scientific literature. In *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2024*, pages 8–23. World Scientific, 2023.

[27] National Library of Medicine (US). Umls knowledge sources, 2024. Release 2024AB. Bethesda (MD): National Library of Medicine (US).

[28] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[29] prithivMLmods. Qwen-umls-7b-instruct. https://huggingface.co/prithivMLmods/Qwen-UMLS-7B-Instruct, 2025. Hugging Face model card. License: CreativeML OpenRAIL-M. Accessed: 2025-07-26.

[30] P. Qiu, C. Wu, X. Zhang, W. Lin, H. Wang, Y. Zhang, Y. Wang, and W. Xie. Towards building multilingual language model for medicine. *Nature Communications*, 15(1):8384, 2024.

[31] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[32] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[33] C. Shu, B. Chen, F. Liu, Z. Fu, E. Shareghi, and N. Collier. Visual med-alpaca: A parameter-efficient biomedical llm with visual capabilities, 2023.

[34] C. Sinsky, L. Colligan, L. Li, M. Prgomet, S. Reynolds, L. Goeders, J. Westbrook, M. Tutty, and G. Blike. Allocation of physician time in ambulatory practice: a time and motion study in 4 specialties. *Annals of internal medicine*, 165(11):753–760, 2016.

[35] L. Soldaini and N. Goharian. Quickumls: a fast, unsupervised approach for medical concept extraction. In *MedIR workshop, sigir*, pages 1–4, 2016.

[36] L. P.-Y. Ting, H.-P. Chen, A.-S. Liu, C.-Y. Yeh, P.-L. Chen, and K.-T. Chuang. Early detection of patient deterioration from real-time wearable monitoring system. *arXiv preprint arXiv:2505.01305*, 2025.

[37] L. P.-Y. Ting, Z. Tan, H.-P. Chen, C.-T. Li, P.-L. Chen, K.-T. Chuang, and H. Liu. Cand: Cross-domain ambiguity inference for early detecting nuanced illness deterioration. *arXiv preprint arXiv:2501.16365*, 2025.

[38] G. Wang, G. Yang, Z. Du, L. Fan, and X. Li. Clinicalgpt: large language models finetuned with diverse medical data and comprehensive evaluation. *arXiv preprint arXiv:2306.09968*, 2023.

[39] Y. Wen, Z. Wang, and J. Sun. Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models. *arXiv preprint arXiv:2308.09729*, 2023.

[40] C. Y. Williams, J. Bains, T. Tang, K. Patel, A. N. Lucas, F. Chen, B. Y. Miao, A. J. Butte, and A. E. Kornblith. Evaluating large language models for drafting emergency department discharge summaries. *medRxiv*, 2024.

[41] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.

[42] R. J. Williams and J. Peng. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268, 1991.

[43] C. Wu, W. Lin, X. Zhang, Y. Zhang, W. Xie, and Y. Wang. Pmc-llama: toward building open-source language models for medicine. *Journal of the American Medical Informatics Association*, 31(9):1833–1843, 2024.

[44] H. Wu, P. Boulenger, A. Faure, B. Céspedes, F. Boukil, N. Morel, Z. Chen, and A. Bosselut. Epfl-make at “discharge me!”: An llm system for automatically generating discharge summaries of clinical electronic health record. In *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*, pages 696–711, 2024.

[45] J. Wu, J. Zhu, Y. Qi, J. Chen, M. Xu, F. Menolascina, and V. Grau. Medical graph rag: Towards safe medical large language model via graph retrieval-augmented generation. *arXiv preprint arXiv:2408.04187*, 2024.

[46] G. Xiong, Q. Jin, Z. Lu, and A. Zhang. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 6233–6251, 2024.

[47] G. Xiong, Q. Jin, X. Wang, M. Zhang, Z. Lu, and A. Zhang. Improving retrieval-augmented generation in medicine with iterative follow-up questions. In *Biocomputing 2025: Proceedings of the Pacific Symposium*, pages 199–214. World Scientific, 2024.

[48] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, et al. Qwen2. 5 technical report. *arXiv e-prints*, pages arXiv–2412, 2024.

[49] D. Yang, J. Wei, D. Xiao, S. Wang, T. Wu, G. Li, M. Li, S. Wang, J. Chen, Y. Jiang, et al. Pediatricsgpt: Large language models as chinese medical assistants for pediatric applications. *Advances in Neural Information Processing Systems*, 37:138632–138662, 2024.

[50] Z. Yang, A. Mitra, S. Kwon, and H. Yu. Clinicalmamba: A generative clinical language model on longitudinal clinical notes. In *ClinicalNLP@ NAACL*, 2024.

[51] C. Yin, B. Qian, J. Wei, X. Li, X. Zhang, Y. Li, and Q. Zheng. Automatic generation of medical imaging diagnostic report with hierarchical recurrent neural network. In *2019 IEEE international conference on data mining (ICDM)*, pages 728–737. IEEE, 2019.

[52] C. Zakka, R. Shad, A. Chaurasia, A. R. Dalal, J. L. Kim, M. Moor, R. Fong, C. Phillips, K. Alexander, E. Ashley, et al. Almanac—retrieval-augmented language models for clinical medicine. *Nejm ai*, 1(2):AIoa2300068, 2024.

[53] H. Zhang, J. Chen, F. Jiang, F. Yu, Z. Chen, J. Li, G. Chen, X. Wu, Z. Zhang, Q. Xiao, et al. Huatuogpt, towards taming language model to be a doctor. *arXiv preprint arXiv:2305.15075*, 2023.

[54] J. Zhang, X. Wang, W. Ren, L. Jiang, D. Wang, and K. Liu. Ratt: A thought structure for coherent and correct llm reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26733–26741, 2025.

[55] C. Zhao, Z. Tan, P. Ma, D. Li, B. Jiang, Y. Wang, Y. Yang, and H. Liu. Is chain-of-thought reasoning of llms a mirage? a data distribution lens. *arXiv preprint arXiv:2508.01191*, 2025.

A Ethical Considerations

The MIMIC dataset and UMLS used in this study are publicly available resources. Authorized access to these datasets was obtained in accordance with their respective data use agreements and license terms, with the required training completed for the MIMIC dataset. All MIMIC-IV clinical notes are fully de-identified and compliant with the Health Insurance Portability and Accountability Act (HIPAA). We do not attempt in any way to identify or deanonymize any individuals in the data during our research.

Moreover, the medical professionals involved in the human evaluation possessed authorized access to the MIMIC dataset and have completed the required training. They were appropriately compensated for their contributions to the study.

We emphasize that this study is intended as a supportive tool designed to assist, rather than replace, physician-led analysis.

B Related Work

B.1 Medical-Specialized LLMs

A growing number of medical-specialized LLMs have been pre-trained on clinical corpora, including Meditron [4], ClinicalGPT [38], HuatuoGPT [53], PediatricsGPT [49], ClinicalMamba [50], BioMistral [17], PMC-LLaMA [43], and MMed-Llama3 [30]. These models improve fluency and factuality on tasks such as ICD coding and short-form clinical QA.

B.2 Retrieval-Augmented Generation in Medical

Retrieval-augmented generation (RAG) techniques play a predominant role in the medical domain by enhancing clinical text generation [21, 52, 26, 46, 47, 45]. Recent studies have incorporated knowledge graph (KG) retrieval to guide LLMs toward concise clinical answers. For instance, MindMap [39] and DR.KNOWS [8] retrieve relevant KG triples or paths to prompt the model. Most focus on questions that have direct answers in a single document or involve selecting limited answer options and short-form outputs such as diagnostic options, probable diseases, or drug recommendations.

B.3 Clinical Note Generation

Other efforts focus on distinct settings, such as summarizing doctor–patient dialogues [1] or generating radiology reports [13, 23, 51] from X-ray images.

Although the above approaches achieve strong performance within their respective settings, they mainly focus on generating short-form outputs. A few studies have explored the generation of long-form discharge summaries [20, 44, 40, 7], but these efforts typically rely on rich in-hospital data, such as progress notes or complete EHRs, that only become available after a prolonged hospital stay. By contrast, we tackle a more challenging scenario of generating long-form discharge instructions using only pre-admission data and design an RL-based retriever over the medical knowledge graph to augment LLM generation.

C Implementation Details

C.1 Data Examples and Statistics

In this paper, we define pre-admission information as allergies, chief complaints, and the history of present illness (HPI). We use this information to infer patients' discharge instructions, as shown in Table 3. Also, Table 4 summarizes data statistics.

Table 3: Example template of the pre-admission information (allergies, chief complaint and history of present illness) and the discharge instruction.

Pre-Admission Information
Allergies*: Not Known* ; ***Chief Complaint****: Fatigue*; ***History of Present Illness****: ___ with history of [...]*
Discharge Instruction
You were admitted because of [...] it was determined that this was due to worsening of your [...]. *During your hospital stay, you were treated with [...]. We also started you on a new medication, [...].* *Please discontinue your [...] after discharge.*

Table 4: Statistics of the medical KG and selected discharge instructions. "Std." denotes the Standard Deviation, and "TTR"represents the Type-Token Ratio.

Medical KG		**Discharge Instructions**	
#Concepts	443K	Avg. #Words	106.9
#Relation	107	Std. #Words	59.99
#Clusters	15	Avg. TTR	0.7

C.2 Hyperparameter Settings

For model training, the maximum number of retrieval steps is set to 5, and the embedding dimension is 768. We train the model for 500 epochs with a batch size of 48. The discount factor (γ in Eq. 9) is set to 0.1, and the weight λ (Eq. 7) is set to 10. The number of retrieval processes per sample (G in Eq. 9) and the number of reasoning paths prompted to the LLM are both set to 10.

C.3 Prompt of *ReinRAG*

ReinRAG **Prompt**

```
You are a doctor tasked with generating discharge instructions for patients. You
are equipped with a medical knowledge graph. Always provide clear, actionable
advice and explain medical terms for patient understanding.

Below provides the [EXAMPLE PATIENT CONDITION], [EXAMPLE RETRIEVED REASONING
PATHS] from the medical knowledge graph, and the corresponding [EXAMPLE DISCHARGE
INSTRUCTIONS]. Please use this example as a guide to generate [NEW DISCHARGE
INSTRUCTIONS] for the new patient based on the provided [NEW PATIENT CONDITION]
and [NEW RETRIEVED REASONING PATHS] from the knowledge graph.

Note that the path format of both the [EXAMPLE RETRIEVED REASONING PATHS] and [NEW
RETRIEVED REASONING PATHS] follows this structure: concept [semantic group] →
```

```
relation → concept [semantic group] → ...

Please write the [NEW DISCHARGE INSTRUCTIONS] in a single, flowing paragraph
format without using separate titles or headings. Address the following aspects:
medications, dietary recommendations, activity level adjustments, and any specific
precautions related to the Allergies, Chief Complaint, and History of Present
Illness, without the greeting sentences. Ensure the [NEW DISCHARGE INSTRUCTIONS]
are clearly structured, with actionable advice and all medical terms explained for
the patient's understanding.


[EXAMPLE PATIENT CONDITION]:
{example_patient_condition}

[EXAMPLE RETRIEVED REASONING PATHS]:
{example_retrieved_reasoning_paths}

[EXAMPLE DISCHARGE INSTRUCTIONS]:
{example_discharge_instructions}

[NEW PATIENT CONDITION]:
{new_patient_condition}

[NEW RETRIEVED REASONING PATHS]:
{new_retrieved_reasoning_paths}

[NEW DISCHARGE INSTRUCTIONS]:
```

D Limitations

While *ReinRAG* shows strong performance, several limitations should be acknowledged. First, although our experiments demonstrate improvements in clinical concept coverage and generation quality, more comprehensive human evaluations by physicians are needed to strengthen performance evaluation. Second, the current fixed-length retrieval in *ReinRAG* may limit adaptability to varying patient complexity. Incorporating adaptive reasoning lengths based on prompt context remains an important direction for future work.