RareSyn: Health Record Synthesis for Rare Disease Diagnosis

Huimin Wang[1,*], **Yutian Zhao**[1,*], **Yefeng Zheng**[2], **Xian Wu**[1,†]
[1]Tencent Jarvis Lab, Shenzhen, China
[2]Medical Artificial Intelligence Lab, Westlake University, Hangzhou, China
[1]{yutianzhao, hmmmwang, kevinxwu}@tencent.com
[2]zhengyefeng@westlake.edu.cn

Abstract

Diagnosis based on Electronic Health Records (EHRs) often struggles with data scarcity and privacy concerns. To address these issues, we introduce RareSyn, an innovative data synthesis approach designed to augment and de-identify EHRs, with a focus on rare diseases. The core insight of RareSyn involves using seed EHRs of rare diseases to recall similar records from both common and rare diseases, and then leveraging Large Language Models to substitute the key medical information (e.g., symptoms or examination details) in these records with information from the knowledge graph, thereby generating new EHRs. We first train a transformer Encoder with contrastive learning to integrate various types of medical knowledge. Then, RareSyn engages in iterative processes of recalling similar EHRs, structuring EHRs, revising EHRs, and generating new EHRs until the produced EHRs achieve extensive coverage of the rare disease knowledge. We assess RareSyn based on its utility for diagnosis modeling, the diversity of medical knowledge it incorporates, and the privacy of the synthesized EHRs. Extensive experiments demonstrate its effectiveness in improving disease diagnosis, enhancing diversity, and maintaining privacy.

1 Introduction

Recent advances in artificial intelligence, particularly in Large Language Models (LLMs), have demonstrated significant promise in the clinical diagnosis of diseases based on Electronic Health Records (EHRs) (Poongodi et al., 2021; Nelson et al., 2022; Zhao et al., 2024b, 2025). However, concerns have been raised regarding their effectiveness when dealing with imbalanced training data. The abundance of data for common diseases contrasts sharply with the scarcity of data for rare diseases, potentially hindering the model's ability to accurately diagnose rare conditions (Chen et al., 2024; Zhao et al., 2024a). Additionally, data security and privacy issues significantly hinder data sharing and the development of AI-assisted diagnosis (Scheibner et al., 2021; Chen et al., 2021). Secure and privacy-preserving data sharing is crucial, especially for rare diseases where data is limited (Hernandez et al., 2022). To address the data deficiency problem for rare diseases, researchers have proposed various methods, including knowledge-guided few-shot learning (Zelin et al., 2024; Zhao et al., 2024b), federated learning (Pati et al., 2022), and LLM-based retrieval-augmented generation (Shyr et al., 2023; Chen et al., 2024). However, none of these approaches produces new rare disease data to overcome data scarcity, balance the training dataset, or facilitate secure data sharing.

Data synthesis, describing a paradigm where generating fully synthetic data serves as an alternative to real data (Gonzales et al., 2023), can potentially address data scarcity and privacy issues. However, the process of synthesizing EHRs that are medical fact accurate, representative of rare disease knowledge, de-identified, and capable of enhancing disease diagnosis performance, presents several challenges: 1) The scarcity of real examples makes accurately capturing the full statistical properties of the data difficult; 2) Any deviation from factual information about rare diseases during synthesis can negatively impact the accuracy of diagnostic models; 3) Ensuring the de-identification of real EHRs during synthesis is a significant task.

To combat data scarcity and enrich rare disease samples, we incorporate knowledge graphs (KG) for disease insights and utilize common disease EHRs for varied templates. To ensure the medical accuracy of synthesized EHRs, we use *imap* (Wang et al., 2024), a data structure that parses plain text into term-value pairs, to highlight key information during synthesis. To de-identify and ensure the utility of the synthesized EHRs for diagnosis,

[*]Equal contribution.
[†]Corresponding author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 12311–12327
November 4-9, 2025

we propose a KG entity weighting method. This method emphasizes the differences between the rare disease KG and EHR templates of common diseases, ensuring that the newly generated EHRs are rare disease-aware and untraceable to real samples. With that in mind, we propose **RareSyn**, a medical knowledge-enhanced EHR synthesis framework for rare disease diagnosis. It seeds with a few rare disease EHRs, recalls similar EHRs from both rare and common diseases as templates, and samples entities from rare disease KG to reshape these templates, thereby generating new rare disease EHRs.

Initially, we train a transformer Encoder with diverse medical knowledge in a unified contrastive learning task. Using some seed rare disease EHRs, we then perform a layered recall process that first identifies the most related diseases and subsequently recalls the top similar EHRs from these diseases. Following this, we structure the recalled EHRs using *imap* to emphasize key information such as symptoms, examinations, and treatments. We then replace the content of the recalled *imap* with entities sampled from the rare disease KG, giving high weight to the differences between the recalled *imap* and the related entities of the KG. Finally, we employ LLMs to rephrase the sampled *imaps*, thereby generating new EHRs for rare diseases. We repeatedly execute the above process until the generated rare disease *imaps* achieve extensive coverage of the KG.

To assess whether the synthesized EHRs are factually correct, representative of the target rare disease, and de-identified, we evaluate RareSyn from three dimensions: 1) Validity and Utility, examining if the synthetic EHRs maintain medical accuracy and improve rare disease diagnosis; 2) Diversity, determining if the synthetic EHRs capture the broad statistical properties of the rare disease; 3) Privacy, ensuring that the synthetic data effectively protect the real EHRs from potential identification.

Our contributions can be outlined as follows:

- To address data scarcity and privacy issues for rare diseases, we propose a new framework, RareSyn, where LLMs and Medical Knowledge Graph work together in an iterative process to synthesize new EHRs for rare diseases.
- To assess synthesized EHRs, we compared them with original data and observed superb results in diagnosis modeling utility, knowledge diversity, and content authenticity.
- To facilitate further research, we released a synthesized rare disease EHR dataset comprising 1,455 records covering 23 rare diseases, based on 397 real clinical EHRs and 100 EHRs from medical exams [1].

2 Related Work

Data synthesis typically involves the generation of data through models or algorithms rather than direct human input (Bauer et al., 2024; Long et al., 2024). As reviewed by (Goyal and Mahmoud, 2024), a variety of machine learning methods have been employed for data synthesis, including GAN-based methods (Xu et al., 2019), VAE-based methods (Kingma, 2013), and large language model based methods (Radford et al., 2019; Brown et al., 2020; Meng et al., 2022; Ge et al., 2024).

In the healthcare domain, Buczak et al. (2010) utilized a data-driven approach to produce synthetic EHRs for exploring questions related to disease outbreaks. (Park et al., 2013) proposed a perturbed Gibbs sampler to generate privacy-preserving patient data. Choi et al. (2017) developed medGAN for EHR synthesis, while Han et al. (2024) introduced a discrete diffusion model for generating tabular EHR data in both unconditional and conditional scenarios. Additionally, Theodorou et al. (2023) presented a hierarchical autoregressive language model for longitudinal EHR generation. Kumichev et al. (2024) developed an LLM-based framework for EHR generation.

Despite these advancements, existing methods do not focus on rare disease data synthesis and struggle to generate realistic, diverse, valid, and de-identified EHR data for rare diseases.

3 Methods

RareSyn's core strength lies in its use of LLMs to generate new EHRs for rare diseases, guided by common disease EHR templates and insights from a rare disease medical KG. As illustrated in Figure 1, RareSyn begins with a transformer-based Encoder trained with contrastive learning to integrate various medical knowledge into a unified task. Following Zhao et al. (2024b), we utilize disease-related triples from the medical KG, multiple-choice medical license exam data, and EHR data during training.

We then continue the following process for EHR synthesizing: 1) Layered Recalling of Similar

[1]The dataset will be released at publication.

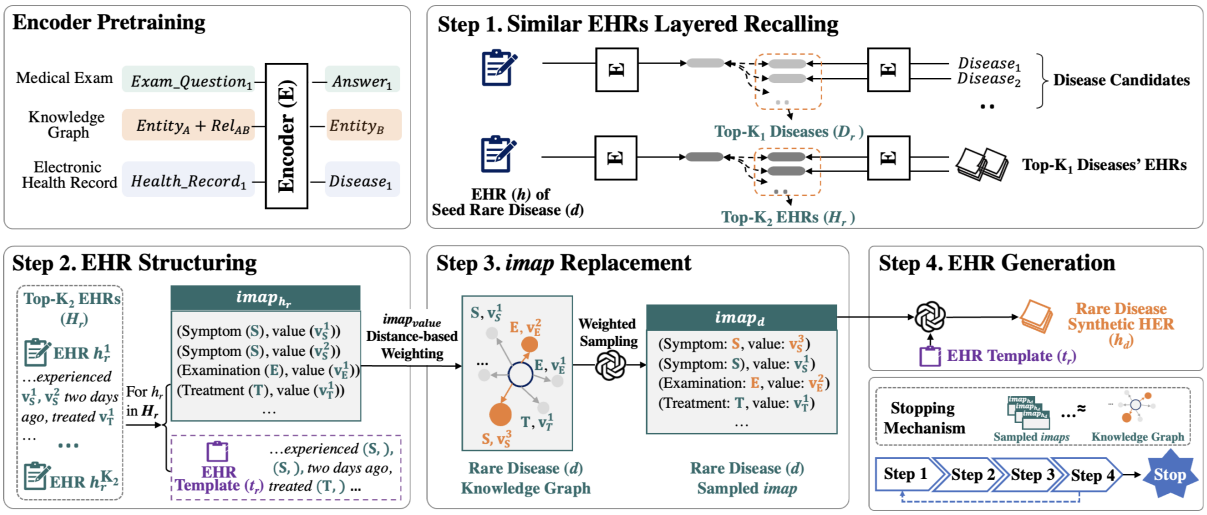

Figure 1: Overview of RareSyn. Initially, we train a transformer Encoder with contrastive learning to integrate various types of medical knowledge. The process then involves rounds of similar EHRs recalling, EHR structuring by *imap*, *imap* replacement, and new EHRs generation to complete the EHR synthesis for rare diseases. A detailed case study and synthetic EHR example are presented in Appendix E and F.

EHRs. Starting with a seed EHR h for a rare disease d, we identify the top K_1 related diseases and retrieve the top K_2 similar EHRs (H_r) for these diseases using the Encoder model; 2) EHR Structuring. For each EHR h_r in H_r, we extract key information from the h_r using the *imap*, resulting in $imap_{h_r}$. Then we mask the values of h_r to create an EHR template t_r; 3) *imap* Replacement. An algorithm is designed to weight the entities in the KG of d. We replace the values in $imap_{h_r}$ by sampling from these weighted entities using a LLM, creating a new *imap* for d, denoted as $imap_d$; and 4) EHR Generation. The $imap_d$ is rephrased using LLMs, guided by the EHR template t_r, to produce readable EHRs. These steps are repeated until the generated rare disease *imaps* achieve comprehensive coverage of the KG for d.

3.1 Layered Recalling of Similar EHRs

RareSyn leverages disease EHR templates and a rare disease knowledge graph to ensure both structural and contextual validation in the synthesized records. Relying solely on rare disease EHRs as templates can result in identical records to the original ones, potentially leading to real EHR leakage and posing privacy issues. To enhance the diversity of synthesized data in a de-identified form, we incorporate common disease EHRs into the process. Starting with a real rare disease EHR h as a seed, we recall similar EHRs from both common and rare diseases. However, directly recalling EHRs can sometimes introduce records that are entirely different from the target diseases, potentially misleading the training process for disease diagnosis. For instance, in our preliminary validation experiments, directly recalling EHRs for "Renal Tuberculosis" resulted in EHRs belonging to AIDS, which has very different clinical notes from "Renal Tuberculosis." Using such templates could negatively impact the diagnosis modeling for "Renal Tuberculosis." Moreover, template EHRs play a crucial role in the *imap* replacement by helping to weight the KG entities, thereby ensuring the utility of the diagnosis task (see section 3.2). However, using template EHRs from too different diseases may reduce their utility for the target disease diagnosis.

To address this, we apply a layered recalling mechanism to retrieve EHRs from similar diseases. Specifically, we use a pre-trained Encoder to encode both the seed EHR h and each disease candidate. As shown in lines 4-8 in Algorithm 1, each disease candidate's representation is compared with the seed EHR's representation, and the diseases with the top K_1 largest cosine similarities are selected as the recalled diseases, denoted as D_r. We next filter the EHRs belonging to D_r to narrow down the candidate set. We then use the Encoder to obtain representations for these EHRs. Each EHR's representation is compared with that of the seed EHR, and the top K_2 EHRs with the highest cosine similarities are selected as the recalled EHRs, denoted as H_r.

3.2 EHR Structuring

In this phrase, to ensure RareSyn produces factually correct EHRs, we extract key information on symptoms, examination, and treatment, and parse the EHRs using *imap*, a data structure that converts plain text into term-value pairs. For example, consider an EHR for "Renal Tuberculosis" that states: "Male, 56, with a 2-week history of low-grade fever and a cough with sputum for 1 month. Chest X-ray reveals irregular patchy shadows and thin-walled cavities in the right lower lobe." The $imap_{h_r}$ for this case is extracted as the following term-value pairs: (Symptoms: 2-week low-grade fever), (Symptoms: Cough with sputum for 1 month), and (Examination: Irregular patchy shadows and thin-walled cavities in the right lower lobe).

Specifically, for a recalled EHR h_r, we parse it using *imap*, denoted as $imap_{h_r}$, focusing on the three dimensions mentioned. We then mask the corresponding term's values within $imap_{h_r}$, creating an EHR template t_r (as illustrated in lines 10-11 of Algorithm 1).

3.3 *imap* Replacement

We then reshape the *imaps* with a medical KG to inject insights about rare diseases, thereby enhancing diagnostic modeling. Specifically, we scan each term-value pair of the extracted *imap* and aim to replace its value by sampling from the corresponding entities in the KG. These entities are obtained from relationships defined by the term's name with the rare disease. For example, for the EHR described in section 3.2, for the term-value pair, (Symptom: Cough with sputum for 1 month), we replace the value by sampling from the entities identified through the relationship between symptoms and the target disease "Renal Tuberculosis" in the KG. However, performing random replacements poses the issue that many diseases share similar symptoms. As a result, there is a risk that all replacements are symptoms common to multiple diseases, rendering the synthesized *imap* ineffective for diagnostic training for the target rare disease. To address this problem, we introduce a weighted sampling mechanism that emphasizes the differences between the KG and the recalled *imaps* for effective and de-identified new EHRs.

Suppose the target disease to be synthesized is d, and the recalled disease EHR is h_r, and its *imap* is formulated as a set of term-value pairs $\{(t_{i_t}, v_{i_t})\}$, where t denotes the terms of Symptom, Examination, and Treatment, respectively. $i_t \in \{1, \ldots, N_t\}$ is the index of the term-value pairs of term t with a maximum number N_t. For a term t_i and its related values V_i, we identify the triplets in the KG that satisfied head entity is d and the relationship is t_i, their tail entities are our candidates for replacing the values of the term t_i. Then, as illustrated in lines 12-16 in Algorithm 1, we calculate the weight for the tail entities. For a tail entity, e, we use the Encoder to calculate the cosine similarities between the values $v \in V_i$ and entity e. The similarity is denoted as $S(v, e)$. To emphasize the differences, the weight of e should be the inverse of $S(v, e)$, with a very small ϵ added for exploration. Additionally, to further improve diversity and reduce repeated sampling, we add an inverse term to the current entity sampled numbers. The weight of sampling the entity e is then formulated as:

$$W(e) = \frac{r(e)}{\sum_{e \in E} r(e)}, \tag{1}$$

where the rating $r(e)$ is given by:

$$r(e) = \log\left(\frac{N}{N_e}\right) + \frac{1}{\max_{v \in V_i}(S(v, e))} + \epsilon, \tag{2}$$

where N is the total number of entities that have been sampled, N_e is the number of times the entity ent has been sampled, and ϵ is a small constant added to ensure exploration. The term $\max_{v \in V_t}(S(v, e))$ represents the maximum cosine similarity between the values in V_i and the entity e.

The weighted KG highlights the different entities that distinguish $imap_{h_r}$ from the KG of disease d, where entities with larger weights are expected to be sampled more frequently. After assigning weights to the KG entities for the target disease d, to effectively instruct the LLM to focus on entities with high weights, we first sample entities from the weighted KG based on their weights calculated by Equation 1. We ensure that the sampled entities include those related to symptoms, examinations, and treatments. We then employ the GPT-4 model (Achiam et al., 2023) to select from these sampled entities and replace the values in the *imap* of the recalled EHR h_r with them, thereby generating a new *imap* for the target rare disease d, denoted as $imap_d$. This process is illustrated in lines 17-19 of Algorithm 1. A detailed prompt for this process is presented in Table 6.

3.4 Rare Disease EHR Generation

In the previous stages, we structured the recalled EHR into the $imap_{h_r}$ and template t_r, and replaced the $imap_{h_r}$'s values with the target rare disease's weighted knowledge graph to produce the new *imap* for the target disease d as $imap_d$. In this stage, we aim to instruct the LLM to generate new EHR text for the rare disease d based on the $imap_d$ and the template t_r.

As shown in Figure 1, we make full use of the $imap_d$, transformed by the weighted KG, in terms of symptom, examination, and treatment for disease d. This is integrated with the EHR templates t_r of related diseases to guide the LLM in filling in the masked content of t_r concerning symptom, examination, and treatment for d. We then feed the $imap_d$, which contains key diagnostic insights of the rare disease d, along with the template t_r to the LLM. Subsequently, we instruct the LLM to select the appropriate entities to fill in all the masks in the template t_r, thereby generating a new EHR h_d for the rare disease d in natural language form. The EHR generation process is detailed in lines 21-22 of Algorithm 1, and the prompt for this process is presented in Table 7.

3.5 Stopping Mechanism

As illustrated in Figure 1, we continue the four steps outlined above: recalling the related diseases and similar EHRs to obtain the EHR h_r, structuring the EHR h_r to get $imap_{h_r}$ and the template t_r, replacing $imap_{h_r}$ to create $imap_d$, and generating the final EHR. This process results in the final synthetic EHR for the rare disease.

We expect to synthesize EHRs enriched with insights from the rare disease KG to enhance rare disease diagnosis. To fully leverage the KG information and improve data synthesis efficiency, we propose the entity weighting mechanism for efficient entity utilization. Once all relevant entities are integrated into the synthetic data, we can conclude the iterative synthesis process described above. To assess the synthetic data's coverage of the rare disease KG, we introduce a metric that measures the proportion of sampled entities in the KG relative to the total number of entities, as follows:

$$\beta(d) = \frac{\sum_{e \in G(d)} \mathbb{I}_{N_e > 0}}{U}, \tag{3}$$

where N_e represents the number of times entity e has been sampled, and $G(d)$ denotes the sub-KG for disease d in terms of symptom, examination, and treatment. The indicator function $\mathbb{I}$ equals 1 if $N_e > 0$ and 0 otherwise. U is the total number of entities in $G(d)$.

The loop "$for\ h_r \in H_r$" in Algorithm 1 can generate multiple EHRs as instructed in the prompt, but we set it to produce one per iteration. The total number of EHRs depends on β and the threshold (we set as 0.98). Specifically, we generated 1,330 rare synthetic EHRs for JARVIS-D and 125 for JarvisD2 (see Appendix Table 4).

Algorithm 1 EHR Synthesizing Algorithm

Require: Target rare disease d, its KG and related entities number U, all EHRs H, K_1, K_2, N, and ϵ
Ensure: Synthesized EHRs $\hat{H}_d$ for d
1: Init the Encoder model M, set $\beta, N_e = 0$
2: **while** $\beta <= 0.98$ **do**
3: $N = N + 1$
4: *# layered recalling of similar EHRs*
5: Randomly Select seed a EHR h of disease d
6: Use M to get K_1 diseases related to h, as D_r
7: Use M to get K_2 EHRs to h diagnosed in D_r, as H_r
8: **for** h_r in H_r **do**
9: *# EHR structuring*
10: Use LLMs to structure h_r, as $imap_{h_r}$
11: Mask $imap_{h_r}$ value in H_r to create template t_r.
12: *# KG Entity Weighting*
13: **for** Entity e in KG with head $= d$ and relationship $\in$ {Symptom, Examination, Treatment} **do**
14: $N_e += 1$
15: Use M to compute e's similarity with $imap_{h_r}$'s values, find max, and get e's weight via eq. 1, 2
16: **end for**
17: *# imap Replacement*
18: Sample entities E_w based on weighted KG
19: Instruct LLM to replace $imap_{h_r}$'s values with E_w to get *imap* for d, as $imap_d$
20: *# EHRs Generation*
21: Guide LLM to synthesize EHR h_d on $imap_d$ & t_r
22: Update β via equ. 3
23: **end for**
24: **end while**

4 Experimental Settings

4.1 Datasets and Baseline

Medical Knowledge Graph. We used the medical knowledge graph[2] in RareSyn and, following (Zhao et al., 2024b), trained the encoder for disease and EHR retrieval with 2,585 disease-related triples. Each triple consists of two entities and a relationship, in the form (entity$_a$, relation, entity$_b$); for example, (Tuberculosis of kidney, Symptom, Back pain). We also incorporated question-answer pairs from medical licensing exams for training (see Appendix A for details).

[2]https://jarvislab.tencent.com/kg-intro.html

Table 1: This table shows the Micro-F1 scores for rare diseases (on JARVIS-D_{rare} and JarvisD2$_{rare}$) and overall diagnosis (on JARVIS-D and JarvisD2). We compare the results of training with only original EHRs and with additional synthetic rare disease EHRs from MedSyn and RareSyn (ours) across different diagnosis models. GPT-4, DeepSeek-R1, and MedPaLM-2 use in-context learning with either 4 original EHR examples or a mix of 2 original and 2 synthetic examples. All RareSyn instances significantly ($p < 0.05$) outperform Original and MedSyn. The highest F1 score is underlined. The Macro-F1 results are presented in Appendix D.

Methods	JARVIS-D_{rare}			JARVIS-D			JarvisD2$_{rare}$			JarvisD2		
	Original	MedSyn	**RareSyn**	Original	MedSyn	**RareSyn**	Original	MedSyn	**RareSyn**	Original	MedSyn	**RareSyn**
Embedding-Based												
BERT (Devlin et al., 2018)	20.5	84.0	92.4	87.3	89.2	89.9	41.2	71.2	78.8	88.1	89.4	91.9
MedBERT (Ting et al., 2020)	21.2	84.7	93.1	87.7	88.3	91.2	47.5	76.2	80.0	88.5	90.6	92.8
GP (Yang et al., 2022a)	21.5	73.6	88.9	84.6	85.8	87.7	42.5	67.5	77.5	86.4	87.2	89.4
KEPT (Yang et al., 2022b)	23.3	81.2	93.1	86.8	87.0	89.5	45.0	73.8	80.0	87.2	88.5	91.5
MKeCL (Zhao et al., 2024b)	25.0	76.4	93.8	88.6	90.3	91.2	50.0	77.5	81.2	89.8	90.6	92.3
General LLMs												
ChatGLM2-6B (GLM et al., 2024)	75.0	83.3	95.5	90.9	92.3	92.5	87.5	90.0	91.2	91.1	92.3	92.8
Qwen1.5-7B (Bai et al., 2023)	37.2	78.5	94.8	88.9	89.2	90.4	90.0	93.8	96.2	93.6	94.5	94.5
GPT-4 (Achiam et al., 2023)	27.8	43.1	44.1	46.6	46.8	48.3	95.0	95.0	95.0	96.6	96.6	97.0
DeepSeek-R1 (Guo et al., 2025)	96.9	97.6	97.6	96.2	96.4	96.4	98.8	98.8	98.8	97.4	97.4	98.3
Specialized LLMs												
HuatuoGPT2-7B (Zhang et al., 2023)	69.1	78.1	94.8	89.2	91.8	92.1	91.2	93.8	95.0	94.0	94.0	94.9
MedPaLM-2 (Singhal et al., 2025)	29.2	38.2	43.1	45.3	46.1	46.1	85.0	86.2	87.5	91.5	91.5	92.8

Electronic Health Records. We evaluated RareSyn using two datasets: JARVIS-D (Zhao et al., 2024b) and JarvisD2 (Zhao et al., 2025). JARVIS-D contains 12,776 EHRs from five hospitals, covering 193 diseases with patient demographics, complaints, exams, and treatments. We split it into JARVIS-D_{common} and JARVIS-D_{rare}, the latter comprising the rarest 9.3% of diseases (3% of EHRs, 18 diseases). JarvisD2 includes 10,953 diagnosis question-answer pairs from CMExam, CMB, and MedBench, spanning 4,949 diseases. After filtering for diseases with at least 20 questions, we obtained 929 pairs across 36 diseases. Using GPT-4, we extracted EHR-diagnosis pairs and further split JarvisD2 into common and rare subsets, with JarvisD2$_{rare}$ containing the five rarest diseases. As real-world rare disease data are extremely scarce and subject to privacy restrictions, our rare disease dataset comprises long-tail disease classes—an approach widely adopted as a proxy in machine learning research (Li et al., 2019; Zhang et al., 2024).

For both JARVIS-D_{rare} and JarvisD2$_{rare}$, we used 16 EHRs per rare disease for testing, with the remainder for training, ensuring thorough evaluation of synthetic EHRs' utility. For JARVIS-D_{common} and JarvisD2$_{common}$, we split them 80-20% into training and testing datasets. All training datasets were used during the Encoder pretraining stage. More details are in Appendix A.

Baseline. We compared MedSyn (Kumichev et al., 2024), which uses LLMs to generate synthetic EHRs by sampling medical contexts from external knowledge bases. Due to the limited rare disease EHR data, other synthesis methods like MedGAN were not applicable.

4.2 Implementation

We trained a BERT-based Encoder for disease and EHRs recalling with contrastive learning on question-answer pairs derived from medical knowledge graphs, medical licensing exams, and EHRs. For similar EHRs layered recalling, we selected one rare disease EHR as a seed, then retrieved the top 5 related diseases and top 5 EHRs. GPT-4 was prompted to generate the *imap* for these EHRs, following Wang et al. (2024). After weighting KG entities, GPT-4 replaced the *imap* to produce the final synthetic EHRs. This process was repeated until the synthetic EHRs' *imaps* fully covered the rare disease KG and matched the average size of common disease EHRs (about 70 for JARVIS-D_{rare} and 25 for JarvisD2$_{rare}$). Example prompts and dataset details are in Appendices G and A.

We assessed the utility of RareSyn-generated synthetic EHRs by training various models on a multi-class disease diagnosis task, including embedding-based models (BERT, MedBERT, GP, KEPT, MKeCL), general LLMs (ChatGLM2-6B, Qwen1.5-7B, GPT-4), and specialized LLMs (HuatuoGPT2-7B, MedPaLM-2). We selected these models based on the state-of-the-art methods for disease diagnosis task. Due to resource limits, only smaller models (6B/7B) were fine-tuned, while larger models (e.g., GPT-4, DeepSeek) used in-context learning. All models were trained in two settings: (1) with original EHRs only, and (2) with both original and synthetic EHRs. Additional

details are in Appendix B.

5 Main Results and Analysis

We conducted extensive evaluations of RareSyn's effectiveness in rare disease diagnosis, along with analyses of its medical factual correctness (Validity), breadth of medical knowledge (Diversity), and de-identification capability (Privacy).

5.1 Main Results

Table 1 presents the Micro-F1 scores for disease diagnosis when training different models using only original EHRs, as well as with additional synthetic rare disease EHRs generated by RareSyne (Ours) or MedSyn, on both JARVIS-D and JarvisD2. The results show that RareSyne consistently outperformed MedSyn across all models and datasets. By using a two-tier selection of real EHR templates and KG sampling that highlights distinguishing features, RareSyne generates more realistic and diverse rare disease notes. In contrast, MedSyn's limited template diversity and less effective sampling often miss key symptoms, resulting in less accurate synthetic data. Furthermore, all models showed significant improvements in both rare disease and overall disease diagnosis Micro-F1 scores when synthetic EHRs were incorporated into the training dataset (evidenced by the comparison between Original and RareSyn/MedSyn).

Comparing the results across different diagnostic models, we found that the improvements for GPT-4, MedPaLM-2, and DeepSeek-R1 were modest, likely because they were trained with in-context learning. Notably, the exceptionally high F1 scores of ChatGLM2, HuatuoGPT2, and DeepSeek-R1 on JARVIS-D_{rare} suggest possible data leakage. Similarly, all LLMs performed much better on JarvisD2, likely due to its open-source data being included in pre-training.

During the similar EHRs layered recall stage, our first recalling diseases as constraints prevents noisy EHR templates from unrelated diseases. This advantage is demonstrated by the superior performance of "RareSyn" over "w/o layered recall" in Figure 2(a). For example, as shown in Figure 2(b), using Renal Tuberculosis EHRs as seeds can recall EHRs from diseases like AIDS, since patients may share a history of tuberculosis. First recalling similar diseases and then recalling EHRs within those disease categories ensures the recalled EHRs are all renal-related. Moreover, weighted *imap* sampling ensures the synthetic EHR is distinct from its templates. As shown in Figure 2(a), this improves disease diagnosis accuracy by 10.7% on average compared to without weighting. Furthermore, using common disease EHRs as templates in RareSyn can diversify the expression of synthetic EHRs, especially when original EHRs are scarce. As shown in Figure 2(a), omitting common EHR templates ("w/o Common EHRs") reduces accuracy on JARVIS-D_{rare} by 5.5%, 4.5%, and 2.7% for MKeCL, Qwen1.5, and HuatuoGPT2, respectively.

5.2 Analysis

Validity We conducted two tests with the assistance of three medical experts to manually verify the validity of the synthetic EHRs. Firstly, we randomly selected 20 synthetic EHRs for each rare disease in JARVIS-D_{rare} and JarvisD2$_{rare}$ and asked the experts to verify if each synthetic EHR was medically accurate and corresponded to the target rare disease. The average accuracy across all diseases and experts was around 97% for both datasets. Secondly, we created 20 pairs of real and synthetic EHRs for each rare disease, using the remaining synthetic EHRs not used in the first test. The experts were then tasked with distinguishing the synthetic EHR in each pair. The average success rate across all experts was approximately 51.6% for JARVIS-D_{rare} and 48.3% for JarvisD2$_{rare}$, indicating that the synthetic EHRs were highly similar to the real ones, making them challenging to differentiate. Results are detailed in Table 2, with human annotation process in Appendix C.

Diversity The diversity of knowledge and template expressions in synthetic EHRs is crucial. As shown in Figure 2(c), diagnostic accuracy improves with increased rare disease KG coverage, but adding more synthetic EHRs after full coverage may slightly reduce performance. To further confirm this, we reduced EHR volume while maintaining full KG coverage and found that performance remained stable, indicating data size has minimal impact once full coverage is achieved.

Figure 3 provides a visualization of EHRs for several common diseases, as well as both original and synthetic EHRs for rare diseases in JARVIS-D. It's evident that the synthetic data for a given rare disease clusters around its original data and is well separated from other diseases. Beyond just overlapping with the primary cluster in the original EHRs, the synthetic Renal Tuberculosis EHRs also create

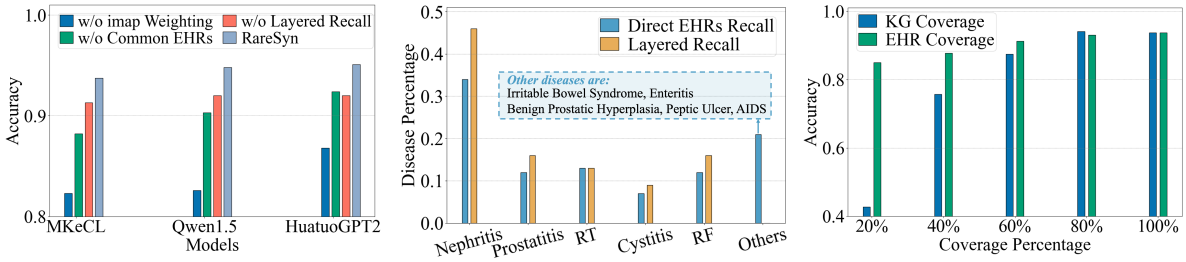

Figure 2: (a) Impact of different components in RareSyn on MKeCL, Qwen1.5, and HuatuoGPT2's rare disease diagnosis accuracy, including *imap* distance-based weighting, layered recalling of similar EHRs, and the use of common disease EHRs as templates for synthesis. (b) Comparison of disease distribution among similar EHRs recalled using "Renal Tuberculosis" EHRs as the seed, through direct EHRs recalling and layered recalling methods. (c) Rare disease diagnostic accuracy of MKeCL under two conditions: 1) trained with synthetic EHRs at different KG coverage levels, and 2) trained with reduced EHR volumes while maintaining full KG coverage. The experiments are performed on JARVIS-D_{rare} (see Appendix D for JarvisD2$_{rare}$ results).

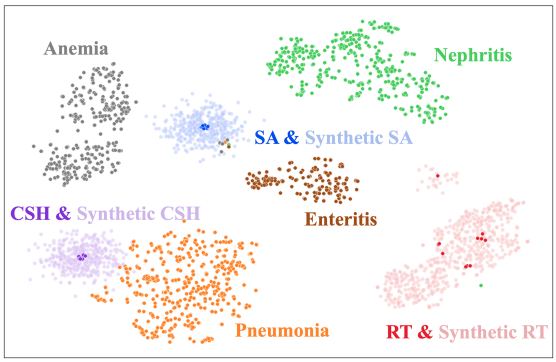

Figure 3: Visualization of EHRs generated by MKeCL using t-SNE. The EHRs include four common diseases (Anemia, Nephritis, Enteritis, Pneumonia), and original and synthetic EHRs for three rare diseases (Renal Tuberculosis (RT), Chronic Subdural Hematoma (CSH), Subphrenic Abscess (SA)) in JARVIS-D.

a cluster around an outlier. This indicates that the process of synthesizing EHRs with diverse medical knowledge not only broadens the information spectrum in rare disease EHRs but also ensures that outliers are given due attention and incorporated during the generation of synthetic data.

Table 2: Validity evaluation of RareSyn. The table shows the accuracy rates of three experts assessing the medical accuracy (Acc.) of 20 sampled synthetic EHRs for each rare disease, and the success rates (SR) of these experts in differentiating between 20 sampled pairs of original and synthetic EHRs per rare disease.

Dataset	Evaluators	Accuracy Test (Acc.)			Identification Test (SR)		
		Min	Max	Avg	Min	Max	Avg
JARVIS-D_{rare}	$Expert_1$	90.0	100.0	**97.2**	45.0	60.0	**53.3**
	$Expert_2$	85.0	100.0	**96.7**	40.0	65.0	**49.4**
	$Expert_3$	90.0	100.0	**97.5**	45.0	65.0	**52.2**
JarvisD2$_{rare}$	$Expert_1$	95.0	100.0	**98.0**	40.0	50.0	**45.0**
	$Expert_2$	90.0	95.0	**94.0**	40.0	55.0	**49.0**
	$Expert_3$	95.0	100.0	**99.0**	45.0	60.0	**51.0**

Table 3: Privacy evaluation of RareSyn. This table shows the minimum and average cosine distances (using BERT) between synthetic and original EHRs. All values are multiplied by 100 for clarity.

Dataset	Training Data	Min Dist	Avg Min Dist
JARVIS-D_{rare}	Original vs Original	3.25	4.25
	Original vs Synthetic	4.18	4.97
JarvisD2$_{rare}$	Original vs Original	3.91	5.11
	Original vs Synthetic	4.52	6.07

Privacy To evaluate the privacy of our synthetic EHRs, we measured the smallest distance in the embedding space between the synthetic and original data in JARVIS-D_{rare} and JarvisD2$_{rare}$. As shown in Table 3, the minimum distance between a synthetic EHR and an original EHR is greater than the smallest distance within the original EHRs themselves for both datasets. Additionally, the average minimum distance between the original and synthetic data groups is slightly higher than the average minimum distance within the original data groups. These larger distances suggest that the synthetic data points are more unique and less similar to the original dataset compared to the similarity among the original data points. The fact that these distances are larger indicates that the synthetic data does not closely mimic specific instances from the original dataset. This effectively demonstrates the synthetic data's ability to maintain privacy, as it reduces the risk of sensitive information being inferred from the synthetic data.

A **case study** on how RareSyn generates synthetic EHRs is presented in Appendix E.

6 Conclusion

To address data scarcity and privacy issues in rare disease diagnostic modeling based on EHRs,

we propose RareSyn, a synthetic data generation method. RareSyn leverages KG for rare disease insights and common disease EHRs for varied templates. It recalls similar EHRs from both common and rare diseases, extracts key information using a special data structure called *imap*, reshapes the *imap* with a novel KG entity-weighted algorithm, and produces new EHRs based on the reshaped *imap* and recalled EHR templates. Extensive experiments demonstrate RareSyn's effectiveness in disease diagnosis improving, medical factual correctness, knowledge diversity, and de-identification capability.

Limitations

We acknowledge three limitations of our study.

Firstly, in biomedical literature, "rare diseases" are typically defined by population prevalence thresholds (e.g., affecting fewer than 1 in 10,000 individuals), and that many conditions in the tail of the JARVIS dataset do not strictly meet this definition. However, due to the extreme scarcity and privacy constraints surrounding real-world rare disease data, it is a widely accepted practice in machine learning research to use long-tail or low-frequency classes as proxies for rare diseases (Li et al., 2019; Zhang et al., 2024). This proxy approach enables reproducible experimentation and benchmarking, and is especially valuable for developing and evaluating methods tailored to the low-sample regime characteristic of true rare diseases.

Secondly, the scope of our study is somewhat narrow, as it only investigates rare disease data synthesizing in Chinese. A potential progression of this research would involve expanding the range of diseases studied and exploring additional language systems.

Thirdly, our fine-tuned baseline LLM models are approximately 7 billion parameters in size, and their results may differ from those of larger models. Due to resource limitations, we were unable to fine-tune larger LLMs for comparison. Future research could extend our experiments by fine-tuning larger LLMs to further validate the superiority of the proposed framework.

Ethics Statement

Our work adheres to the ACL Ethics Policy. This paper aims to highlight the synthesis of electronic health records (EHRs) for rare disease diagnosis, addressing potential issues from improper application of the proposed models in the medical domain. The primary objective is to explore an effective EHR synthesis method using LLMs to alleviate data scarcity and privacy concerns in rare disease diagnosis modeling. However, it is crucial to note that these methods and the synthetic data are not yet ready for real-world medical deployment. A significant concern is the potential for these methods to mislead users about the reasons behind their predictions, which could lead to incorrect decisions and serious implications for patient care and outcomes.

Beyond accuracy and reliability, the ethical considerations of our work include the privacy and security of sensitive medical data. We have enforced rigorous measures to safeguard this information throughout the data collection and utilization process, even when using previously proposed datasets. In conclusion, while our work shows promise for improving disease diagnosis, it is essential to approach its application with caution. We must continue to prioritize ethical considerations of accuracy, transparency, data privacy, and security as we further develop and refine these methods.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. arXiv preprint arXiv:2309.16609.

André Bauer, Simon Trapp, Michael Stenger, Robert Leppich, Samuel Kounev, Mark Leznik, Kyle Chard, and Ian Foster. 2024. Comprehensive exploration of synthetic data generation: A survey. arXiv preprint arXiv:2401.02524.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901.

Anna L Buczak, Steven Babin, and Linda Moniz. 2010. Data-driven approach for creating synthetic electronic medical records. BMC medical informatics and decision making, 10:1–28.

Richard J Chen, Ming Y Lu, Tiffany Y Chen, Drew FK Williamson, and Faisal Mahmood. 2021. Synthetic

data in machine learning for medicine and healthcare. Nature Biomedical Engineering, 5(6):493–497.

Xuanzhong Chen, Xiaohao Mao, Qihan Guo, Lun Wang, Shuyang Zhang, and Ting Chen. 2024. Rarebench: Can llms serve as rare diseases specialists? In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 4850–4861.

Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng Sun. 2017. Generating multi-label discrete patient records using generative adversarial networks. In Machine learning for healthcare conference, pages 286–305. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional Transformers for language understanding. arXiv preprint arXiv:1810.04805.

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas. arXiv preprint arXiv:2406.20094.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, et al. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. arXiv preprint arXiv:2406.12793.

Aldren Gonzales, Guruprabha Guruswamy, and Scott R Smith. 2023. Synthetic data in health care: A narrative review. PLOS Digital Health, 2(1):e0000082.

Mandeep Goyal and Qusay H Mahmoud. 2024. A systematic review of synthetic data generation techniques using generative ai. Electronics, 13(17):3509.

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

Jun Han, Zixiang Chen, Yongqian Li, Yiwen Kou, Eran Halperin, Robert E Tillman, and Quanquan Gu. 2024. Guided discrete diffusion for electronic health record generation. arXiv preprint arXiv:2404.12314.

Mikel Hernandez, Gorka Epelde, Ane Alberdi, Rodrigo Cilla, and Debbie Rankin. 2022. Synthetic data generation for tabular health records: A systematic review. Neurocomputing, 493:28–45.

Diederik P Kingma. 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Gleb Kumichev, Pavel Blinov, Yulia Kuzkina, Vasily Goncharov, Galina Zubkova, Nikolai Zenovkin, Aleksei Goncharov, and Andrey Savchenko. 2024. Medsyn: Llm-based synthetic medical text generation framework. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 215–230. Springer.

Xuedong Li, Yue Wang, Dongwu Wang, Walter Yuan, Dezhong Peng, and Qiaozhu Mei. 2019. Improving rare disease classification using imperfect knowledge graph. In 2019 IEEE International Conference on Healthcare Informatics (ICHI), pages 1–2. IEEE.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On llms-driven synthetic data generation, curation, and evaluation: A survey. arXiv preprint arXiv:2406.15126.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. Advances in Neural Information Processing Systems, 35:462–477.

Charlotte A Nelson, Riley Bove, Atul J Butte, and Sergio E Baranzini. 2022. Embedding electronic health records onto a knowledge network recognizes prodromal features of multiple sclerosis and predicts diagnosis. Journal of the American Medical Informatics Association, 29(3):424–434.

Yubin Park, Joydeep Ghosh, and Mallikarjun Shankar. 2013. Perturbed gibbs samplers for generating large-scale privacy-safe synthetic health data. In 2013 IEEE International Conference on Healthcare Informatics, pages 493–498. IEEE.

Sarthak Pati, Ujjwal Baid, Brandon Edwards, Micah Sheller, Shih-Han Wang, G Anthony Reina, Patrick Foley, Alexey Gruzdev, Deepthi Karkada, Christos Davatzikos, et al. 2022. Federated learning enables big data for rare cancer boundary detection. Nature communications, 13(1):7346.

T Poongodi, D Sumathi, P Suresh, and Balamurugan Balusamy. 2021. Deep learning techniques for electronic health record (ehr) analysis. Bio-inspired Neurocomputing, pages 73–103.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9.

James Scheibner, Jean Louis Raisaro, Juan Ramón Troncoso-Pastoriza, Marcello Ienca, Jacques Fellay, Effy Vayena, and Jean-Pierre Hubaux. 2021. Revolutionizing medical data sharing using advanced privacy-enhancing technologies: technical, legal, and ethical synthesis. Journal of medical Internet research, 23(2):e25120.

Cathy Shyr, Yan Hu, Paul A Harris, and Hua Xu. 2023. Identifying and extracting rare disease phenotypes with large language models. arXiv preprint arXiv:2306.12656.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, et al. 2025. Toward expert-level medical question answering with large language models. Nature Medicine, pages 1–8.

Brandon Theodorou, Cao Xiao, and Jimeng Sun. 2023. Synthesize high-dimensional longitudinal electronic health records via hierarchical autoregressive language model. Nature communications, 14(1):5305.

Liu Ting, Qin Bing, Liu Ming, Xu Ruifeng, Tang Buzhou, and Chen Qingcai. 2020. Pre-training model for Chinese medical text processing PCL MedBERT. PCL blog.

Huimin Wang, Yutian Zhao, Xian Wu, and Yefeng Zheng. 2024. imapscore: Medical fact evaluation made easy. In Findings of the Association for Computational Linguistics ACL 2024, pages 10242–10257.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling tabular data using conditional gan. Advances in neural information processing systems, 32.

Zhichao Yang, Sunjae Kwon, Zonghai Yao, and Hong Yu. 2022a. Multi-label Few-shot ICD Coding as Autoregressive Generation with Prompt. arXiv preprint arXiv:2211.13813.

Zhichao Yang, Shufan Wang, Bhanu Pratap Singh Rawat, Avijit Mitra, and Hong Yu. 2022b. Knowledge Injected Prompt Based Fine-tuning for Multi-label Few-shot ICD Coding. arXiv preprint arXiv:2210.03304.

Charlotte Zelin, Wendy K Chung, Mederic Jeanne, Gongbo Zhang, and Chunhua Weng. 2024. Rare disease diagnosis using knowledge guided retrieval augmentation for chatgpt. Journal of Biomedical Informatics, 157:104702.

Hongbo Zhang, Junying Chen, Feng Jiang, Fei Yu, Zhihong Chen, Jianquan Li, Guiming Chen, Xiangbo Wu, Zhiyi Zhang, Qingying Xiao, et al. 2023. Huatuogpt, towards taming language model to be a doctor. arXiv preprint arXiv:2305.15075.

Tianjiao Zhang, Chaofan Ma, and Yanfeng Wang. 2024. Tracking the rareness of diseases: Improving long-tail medical detection with a calibrated diffusion model. Electronics, 13(23):4693.

Yutian Zhao, Huimin Wang, Yuqi Liu, Wu Suhuang, Xian Wu, and Yefeng Zheng. 2024a. Can LLMs replace clinical doctors? exploring bias in disease diagnosis by large language models. In Findings of the Association for Computational Linguistics: EMNLP 2024, pages 13914–13935, Miami, Florida, USA. Association for Computational Linguistics.

Yutian Zhao, Huimin Wang, Xian Wu, and Yefeng Zheng. 2024b. Mkecl: Medical knowledge-enhanced contrastive learning for few-shot disease diagnosis. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 11394–11404.

Yutian Zhao, Huimin Wang, Yefeng Zheng, and Xian Wu. 2025. A layered debating multi-agent system for similar disease diagnosis. In Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers), pages 539–549.

A Datasets

Medical Licensing Exams. We used 41,626 multiple-choice questions from past Medical Licensing Exams for Encoder pretraining. These questions span six categories of medical knowledge: treatment, lab test, body part, medicine, disease cause, symptom, and others, comprising 33.6%, 23.5%, 1.1%, 5.3%, 5.3%, 9.1%, and 18.6% of the data, respectively. Each exam question was converted into a question-answer pair, with the correct answer forming a positive instance and each incorrect option forming a negative instance. We extracted the EHR descriptions from diagnostic medical examination questions. These questions are meticulously edited and high in information density, ensuring that the clinical text can be definitively diagnosed.

An example is:

Female, young. Suddenly experienced chills, high fever, lower back pain, and symptoms of frequent urination and painful urination for a week. She has no history of similar episodes. Examination: Body temperature 39.4°C, positive percussion pain in the right kidney area, urine protein (+), 20-30 white blood cells/HP, 0-2 white blood cell casts/low power field. What is the most likely diagnosis for this patient?

We can extract the description part as the EHR.

JARVIS-D$_{\text{rare}}$. The tail 18 disease EHRs in JARVIS-D account for 9.3% of all diseases, representing 3% of JARVIS-D. These tail diseases and their corresponding EHR counts are Obsessive-Compulsive Disorder(22), Sigmoid Volvulus(22), Hypopituitarism(22), Rickets(22), Cystitis(22), Esophagitis(21), Hematogenous Pulmonary Abscess(21), Pulmonary Embolism(21), Eclampsia(21), Acute Stress Disorder(21), Periodic Paralysis(20), Uterine Perforation(20), Hypoxic Ischemic Encephalopathy(20), Gonorrhea(20), Dermatomyositis(20), Subphrenic Abscess(20), Chronic Subdural Hematoma(20), and Renal Tuberculosis(20).

JarvisD2$_{\text{rare}}$. Since the original JarvisD2 contains 10,953 disease diagnosis questions covering 4,949 distinct diseases, and most of these diseases have fewer than 3 corresponding questions, we filtered out diseases with at least 20 questions each to create a dataset for our disease diagnosis classification task. The tail 5 disease EHRs account for 13.9% of the diseases, respresenting 10.8% of the filtered JarvisD2. These tail diseases and their corresponding EHR counts are Adenomyosis(20), Ventricular Septal Defect(20), Phenylketonuria(20), Peptic Ulcer(20) and Pulmonary Tuberculosis(20).

Synthetic EHRs. Using EHRs in JarvisD2$_{\text{rare}}$ and JARVIS-D$_{\text{rare}}$ as seeds, we created their corresponding synthetic EHR datasets using RareSyn. More dataset details for JARVIS-D, JarvisD2, and their corresponding original and synthetic rare disease datasets are presented in Table 4.

B Experiment Settings

Implementations In training various models on the disease diagnosis task, we applied the subsequent hyperparameter configurations:

- All embedding-based models were trained with a learning rate of 1×10^{-4}, 100 warm-up steps, a batch size of 16, a maximum sequence length of 256 and a maximum of 100 epochs.
- ChatGLM2-6B, Qwen1.5-7B, and HuatuoGPT2-7B were fully fine-tuned using 8 V100 with deepspeed, ZeRO stage 2, fp16 enabled, a learning rate of 1×10^{-5}, a batch size of 1, gradient accumulation steps 16, and a maximum of 3 epochs.
- For GPT-4, DeepSeek-R1 and MedPaLM-2, we used in-context learning to simulate the training process by providing 4 examples to the model. We compared the results of sampling examples entirely from original EHRs with those that sampled half from original EHRs and half from synthetic EHRs.

C Human Evaluation

The medical experts involved in the validation process were medical students from our partner hospitals. Their participation was voluntary, and they were not compensated for their contributions. We provide detailed human evaluation instructions as following:

Table 4: Dataset details for JARVIS-D, JarvisD2, and their corresponding original and synthetic rare disease datasets.

Dataset	# of Diseases	# of EHRs	EHR Avg Length
JARVIS-D	193	12,776	87.5
JARVIS-D_{rare}	18	397	76.8
JARVIS-$D_{\text{rare synthetic}}$	89	1,330	87.6
JarvisD2	36	929	64.4
JarvisD2$_{\text{rare}}$	5	100	57.5
JarvisD2$_{\text{rare synthetic}}$	5	125	65.3

Annotation Process

Phase 1: Synthetic EHRs' Medical Factual Correctness Verification

- Carefully check the demographics, symptom logic, lab results (with references), and diagnostic disease.

Annotation:

- **Accuracy:**
  - Fully Accurate: No contradictions
  - Partially Accurate: $\leq$2 errors
  - Inaccurate: $>$2 errors
- **Error Marking:**
  - Highlight in red; comment on error type (e.g., Data Contradiction, Temporal Inconsistency) and suggest revisions.
- **Confidence:** 1–5 scale

Phase 2: Disease Diagnosis Verification

- Carefully review the synthetic EHRs and verify whether their diagnoses match the target rare disease.

Annotation:

- Full Match: exactly the same diagnosis
- Partial Match: related disease, e.g., nephritis, acute nephritis
- Mismatch: incorrect diagnosis
- Confidence: 1–5 scale

Synthetic EHR Validation Protocol

Objective: Evaluate synthetic EHRs for accuracy and disease alignment using evidence-based standards.
Steps:

1. **Medical Accuracy:** Assess temporal logic, data consistency, and treatment appropriateness. Highlight errors in red, specify error type and revision, and assign confidence (1–5).
2. **Disease Alignment:**
   - Full Match: All major criteria
   - Partial Match: $\geq$2 minor criteria
   - Mismatch: Provide ICD-11 code
3. **Confidence:** 5 = clear evidence, 3 = needs confirmation, 1 = speculative

D Experiment Results

This section reports further experimental results and analyses of RareSyn. Table 5 summarizes the Macro-F1 scores for both rare disease diagnosis and overall diagnostic performance. Our method outperformed all baseline methods across all datasets and models. We further report analyses of the relationships between the breadth of medical knowledge encapsulated in synthetic EHRs, the percentage of EHRs employed when achieving full knowledge graph coverage, and the diagnostic accuracy of models trained with these synthetic EHRs. Specifically, we explore how the extent of medical knowledge in synthetic EHRs and the proportion of EHRs used upon reaching full knowledge graph coverage can influence the diagnostic accuracy. Experiment results on JarvisD2$_{\text{rare}}$ is depicted in Figure 4.

Moreover, we conduct an ablative study on RareSyn to examine the effects of *imap* distance-based weighted sampling, layered recall of similar

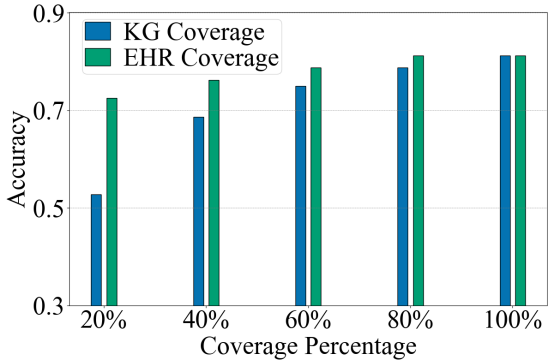

Figure 4: Rare disease diagnostic accuracy of MKeCL on JarvisD2$_{rare}$ when trained with synthetic EHRs of varying KG coverage, and the accuracy when using full KG coverage but with different EHR coverage levels.

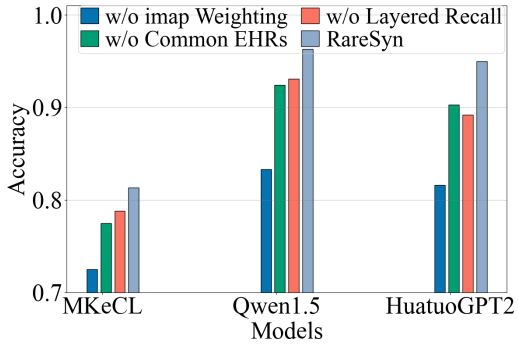

Figure 5: Impact of various RareSyn components on the diagnosis accuracy of MKeCL, Qwen1.5, and HuatuoGPT2 on the JarvisD2$_{rare}$ dataset. These components include *imap* distance-based weighting, layered recall of similar EHRs, and the use of common disease EHRs as templates for synthesis.

EHRs and the use of common disease EHRs as templates for synthesis. The results of these experiments on JarvisD2$_{rare}$ are depicted in Figure 5.

E Case Study

Figure 6 presents a case study illustrating how RareSyn generates synthetic EHRs, specifically for Renal Tuberculosis (RT).

The process begins with a seed RT EHR. Using our trained Encoder, we perform a layered recall of similar EHRs. Initially, we identify diseases most similar to RT. Within this range of diseases, we then recall EHRs that share similarities with our seed RT EHR.

For each similar EHR recalled, we follow steps 2 to 4 in RareSyn to generate a corresponding synthetic RT EHR. For instance, consider a recalled Nephritis EHR. In step 2, the *imap* structuring phase, we extract the *imaps* from this EHR and mask them to create a template.

In step 3, we calculate the weight of each entity in the RT knowledge graph. This is done by comparing them with the *imaps* of the Nephritis EHR and the frequency of their occurrence in existing synthetic RT EHRs. Entities present in the RT knowledge graph but absent in the Nephritis EHR *imaps* are given more weight. For example, 'Normal-sized kidney' and 'Ineffective Anti-Infective Treatment' are key pieces of information that distinguish RT from Nephritis, as Nephritis often leads to enlarged kidneys and can typically be treated with anti-infective therapy.

Finally, in step 4, we use GPT-4 to combine the sampled RT *imaps* and the Nephritis EHR template obtained in step 2. This results in a complete synthetic RT EHR.

F A Synthetic EHR Example

We present an example that demonstrates the process from Seed EHR to Template EHR to Synthetic EHR, using the case of "Renal Tuberculosis." This example includes the original seed EHR, the retrieved template EHR, and the final generated synthetic EHR.

Seed EHR: Female, young. She has been experiencing episodic lower back pain accompanied by frequent urination and urgency for five years. She was found to have hematuria during a routine urine test at the hospital. She was admitted to the hospital due to fever accompanied by lower back pain and painful urination for two days. Examination: T380C. Blood pressure 18.7/2kPa (140/90mmHg). Urine protein (+), red blood cells (++), white blood cells (+++). Kidney ultrasound: right kidney 11cm×5cm×3cm, left kidney 8cm×4cm×2cm. Seed disease: Renal Tuberculosis

Template EHR: Female, young. Sore throat, cough, fever. Noticed red urine two weeks later. Eyelid edema. Urine output 1000ml/24h. Physical examination: no rash on the skin. Blood pressure 150/100mmHg. Laboratory tests: urine protein (++). Red blood cells: 50 60/HP. Blood albumin 329/L. Blood creatinine 123mol/L. Seed disease: Nephritis

Synthetic EHR: Male, young. Recently experiencing lower back pain, decreased urination, no hematuria. Normal body temperature, blood pressure 120/80mmHg. Routine urine test: urine protein (+), urine specific gravity 1.010, red blood

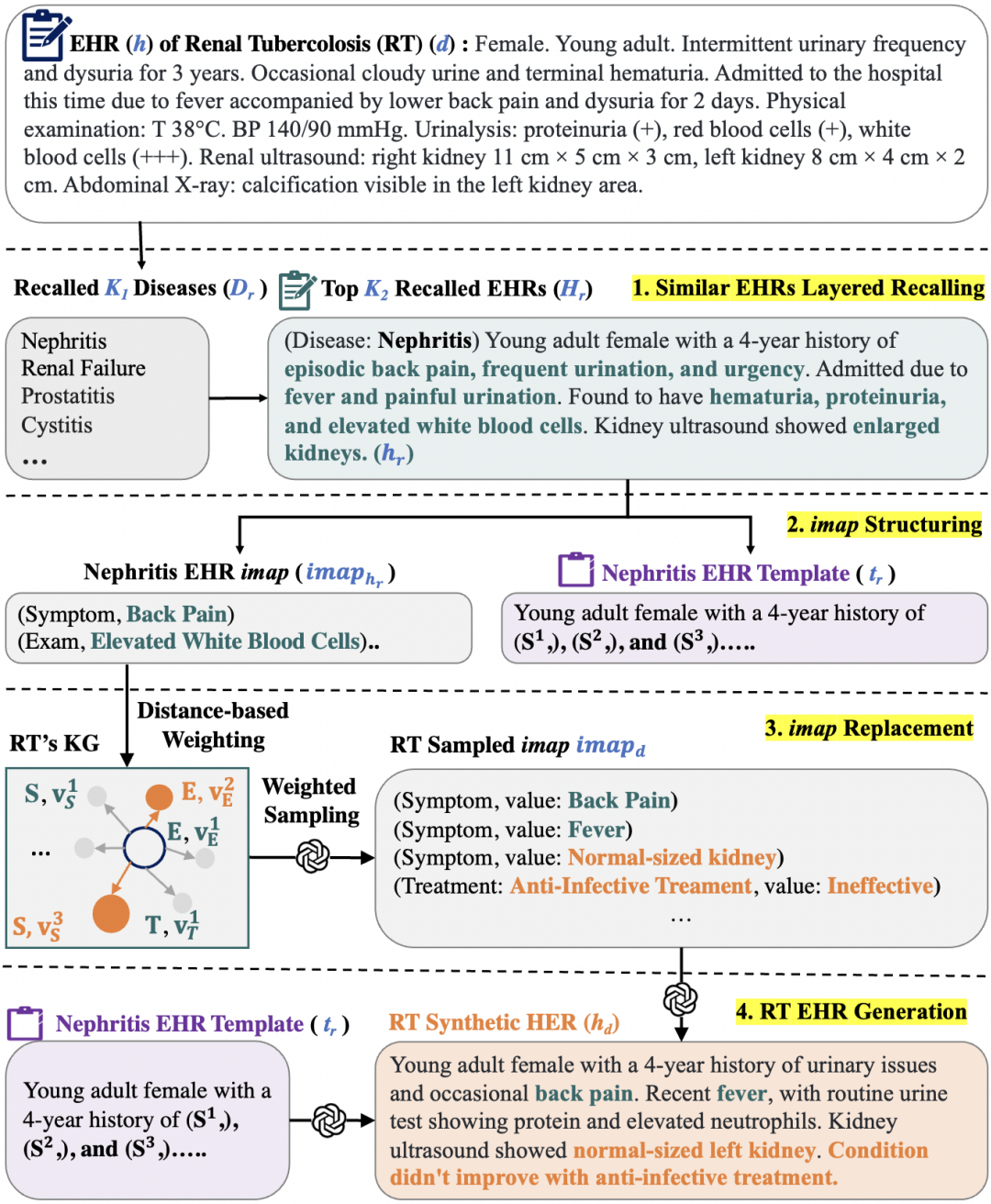

Figure 6: Case study on how RareSyn generates a synthetic Renal Tubercolosis EHR.

Table 5: This table shows the Macro-F1 scores for rare diseases (on JARVIS-D_{rare} and JarvisD2$_{rare}$) and overall diagnosis (on JARVIS-D and JarvisD2). We compare the results of training with only original EHRs and with additional synthetic rare disease EHRs from MedSyn and RareSyn (ours) across different diagnosis models. GPT-4, DeepSeek-R1, and MedPaLM-2 use in-context learning with either 4 original EHR examples or a mix of 2 original and 2 synthetic examples. All RareSyn instances significantly ($p < 0.05$) outperform Original and MedSyn. The highest F1 score is underlined.

Methods	JARVIS-D_{rare}			JARVIS-D			JarvisD2$_{rare}$			JarvisD2		
	Original	MedSyn	**RareSyn**	Original	MedSyn	**RareSyn**	Original	MedSyn	**RareSyn**	Original	MedSyn	**RareSyn**
Embedding-Based												
BERT (Devlin et al., 2018)	20.4	83.9	92.4	84.2	86.8	87.4	41.2	71.1	78.6	87.1	88.6	91.8
MedBERT (Ting et al., 2020)	20.8	84.6	93.1	84.4	86.0	88.8	48.0	76.4	80.3	87.0	89.6	91.8
GP (Yang et al., 2022a)	21.3	73.6	88.8	81.6	82.5	85.4	42.0	67.6	77.2	85.7	85.2	88.6
KEPT (Yang et al., 2022b)	23.4	81.1	93.1	83.2	84.0	86.8	45.3	73.9	79.5	86.1	87.4	91.8
MKeCL (Zhao et al., 2024b)	25.0	76.3	93.7	86.1	88.2	88.9	50.2	77.5	81.8	88.3	89.6	91.6
General LLMs												
ChatGLM2-6B (GLM et al., 2024)	75.0	83.3	95.5	88.7	90.5	90.8	87.3	89.9	91.2	89.7	92.1	92.0
Qwen1.5-7B (Bai et al., 2023)	37.0	78.7	94.7	86.6	86.6	88.2	89.9	93.8	96.2	92.3	93.7	94.2
GPT-4 (Achiam et al., 2023)	27.6	43.2	44.1	41.1	41.9	43.7	94.9	95.0	95.0	96.4	96.4	96.2
DeepSeek-R1 (Guo et al., 2025)	96.8	97.6	97.6	94.9	95.5	95.4	98.7	98.7	98.7	96.6	97.2	98.4
Specialized LLMs												
HuatuoGPT2-7B (Zhang et al., 2023)	68.8	78.1	94.7	86.4	89.5	89.7	91.3	93.8	95.1	93.5	93.5	94.7
MedPaLM-2 (Singhal et al., 2025)	28.9	37.6	42.9	40.1	40.6	41.3	85.0	86.4	87.4	89.4	91.3	91.4

cells (+), white blood cells (++). Kidney ultrasound: right kidney 9cm×4cm×2cm, left kidney 7cm×3cm×2cm. Chest X-ray shows normal heart and lungs. Despite the use of a large amount of antibiotics, the treatment effect is not good. Seed disease: Renal Tuberculosis

G Example Prompts

We provide the details of the prompts used for rare disease EHR *imap* replacement and EHR generation, as presented in Tables 6 and 7.

<Task>: As an expert in the field of rare diseases, specifically [d], your clinical experience is invaluable to us in synthesizing our Electronic Health Record (EHR) data related to [d].
You are given a <Structured EHR> from a different disease, formatted in term-value pairs, as well as a <Knowledge Graph of d>. Your task is to extract related information from this <Knowledge Graph of d> and use it to substitute the values in each term-value pair of the <Structured EHR>. This process will generate a new structured EHR specifically for [d].
<Structured EHR>:
[EHR]
<Knowledge Graph of [d]>:
[KG]
<Output a New Structured EHR for [d]>:

Table 6: Rare disease EHR *imaps* replacement prompt.

<Task>:
As an expert in the field of rare diseases, specifically [d], your clinical experience is invaluable to us in synthesizing our Electronic Health Record (EHR) data related to [d].
<Instructions>:
1. Carefully read the following provided <Knowledge about [d]> and the <EHR template>. Incorporate all the content in <Knowledge about [d]> into the <EHR template> to produce a comprehensive and logical EHR for [d].
2. Ensure that the EHR you produce is reasonable and valid, with no contradictions between gender, age, and symptoms.
3. The completed EHR should contain ample information necessary for the diagnosis of [d].
<Knowledge about [d]>:
[IMAP]
<EHR Template>:
[TEMPLATE]
Please refer to the format of the <EHR Template> and sample specific content from the <Knowledge about [d]> to fill in.
<Output [d] EHR>:

Table 7: Rare disease EHR generation prompt.