

LLMSYN: Generating Synthetic Electronic Health Records Without Patient-Level Data

Yijie Hao

*Department of Computer Science
Emory University*

YHAO49@EMORY.EDU

Joyce C Ho

*Department of Computer Science
Emory University*

JOYCE.C.HO@EMORY.EDU

Huan He

*Department of Mathematics and Statistics
Auburn University*

HUAN.HE@AUBURN.EDU

Abstract

Recent advancements in large language models (LLMs) have shown promise in tasks like question answering, text summarization, and code generation. However, their effectiveness within the healthcare sector remains uncertain. This study investigates LLMs' potential to generating synthetic, structured electronic health records (EHRs). Unfortunately, employing LLMs directly resulted in poor statistical similarity and utility. Although feeding real-world data to LLMs can potentially mitigate this issue, it also raises privacy concerns as this transmits patients' information to the LLM API. To address these challenges and unleash the potential of LLMs for synthetic EHR generation, we present a new generation pipeline called LLMSYN. Our pipeline utilizes only high-level statistical information from datasets and publicly available medical knowledge. Our results demonstrate that the generated EHRs by LLMSYN exhibit improved statistical similarity and utility in downstream tasks and achieve predictive performance comparable to training with real data with minimal privacy risks. Our findings suggest that LLMSYN offers a promising approach to enhance the utility of LLM models in synthetic structured EHR generation.

Keywords: LLM, Inference, EHR, Generation Method

1. Introduction

Foundation models (or large-scale machine learning models trained on massive amounts of data) offer new opportunities for improving diagnostic and predictive capabilities using structured data from electronic health records (EHRs) (Li et al., 2019; Rasmy et al., 2020; Steinberg et al., 2020). However, various concerns over patient privacy need to be accounted for to collect the large quantities of data needed to train these robust feature representations. Recent research suggests a training attack can yield personally identifiable information even on privately trained data (Carlini et al., 2021; Huang et al., 2022). Furthermore, it is challenging for researchers to obtain access to real EHRs to train a deep learning model, let alone ensure there are sufficient records to pre-train a foundation model. One approach to mitigate privacy risks is to de-identify patient information in EHRs using data pertur-

bation and randomization (El Emam et al., 2015; McLachlan et al., 2016). Unfortunately, de-identification can still result in records that are vulnerable to re-identification attacks (Narayanan and Shmatikov, 2008). An alternative approach receiving increasing attention is the creation and dissemination of synthetic datasets that aim to capture many of the complexities of the original data set (e.g., distributions, non-linear relationships, and noise). Synthetic data can yield records robust against re-identification (Giuffrè and Shung, 2023). To support the creation of realistic and synthetic data, generative models have emerged as the key element to capturing the complexity and variability of real-world settings.

Existing works extend classical deep generative models to EHRs, building upon generative adversarial networks (GANs), autoencoders, or diffusion models (Choi et al., 2017; He et al., 2023; Naseer et al., 2023; Torfi and Fox, 2020). Yet these architectures require real EHRs, which can be challenging to obtain access or may generate skewed results when the training sample is small. Another synthetic generation approach, Synthea (Walonoski et al., 2018), avoids access to patient records and instead relies on publicly available statistics, clinical guidelines, and clinical knowledge databases to simulate realistic EHR data (Walonoski et al., 2018). Unfortunately, the data has limited utility, especially for modeling health outcomes and other discriminative tasks (Chen et al., 2019). Given the recent adoption of large language models (LLMs) due to their impressive reasonable capabilities and extensive knowledge from pre-training on a variety of diverse data sources, the question is whether LLMs can generate synthetic EHRs without requiring access to patient records.

We posit that LLMs can bridge the divide between generative models and Synthea. LLMs can generate records that provide utility while still preserving privacy and avoiding the need to train on real EHR data. However, simply prompting an LLM to generate synthetic EHR data does not yield desirable results. Instead, we introduce a novel LLM-based pipeline, LLMSYN, to generate realistic EHRs that closely resemble the original data. Our contributions can be summarized as follows:

- We introduce an EHR generation pipeline that harnesses the capabilities of LLMs. It utilizes publicly available clinical knowledge and solely relies on high-level statistical information from private data. Without transmitting individual patient records to the LLM API, LLMSYN inherently protects patient privacy.
- We assess the performance of different LLMs with LLMSYN on a real-world dataset. Our findings indicate that LLMSYN can generate high-quality EHRs in terms of utility, fidelity, and privacy, underscoring its potential in synthetic EHR generation.

Generalizable Insights about Machine Learning in the Context of Healthcare

In addition to being the first to propose the use of LLM for structured EHR generation, this paper presents a number of generalizable insights including:

- The LLM-based pipeline is capable of generating records that can be used to diversify and enlarge the patient population, thereby enhancing the robustness of the prediction model. This is important for smaller healthcare institutions with limited patient data.
- The quality of the generated synthetic records does not rely on the quality of the underlying EHR data but instead on the capability of the LLM model.

- Our pipeline can be extended to incorporate other tabular data sources (e.g., medications or laboratory results) as well as other relevant knowledge sources (e.g., MedlinePlus, DrugBank, etc.).
- Our pipeline highlights the broader applicability and effectiveness of pre-trained LLMs with appropriate knowledge injection and prompt design.

2. Related Works and Preliminaries

2.1. Synthetic EHR Generation Methods

There are two popular network structures for synthetic EHR generation, GANs (Goodfellow et al., 2014) and Denoising Diffusion models (Ho et al., 2020).

GAN-based Models. GAN uses two neural networks, a generator and a discriminator, where the former is trained to generate synthetic data that can fool the discriminator (trained to detect whether the data is true or generated). Early GAN-based methods, such as MedGAN (Choi et al., 2017), EMR-WGAN (Zhang et al., 2020), and CorGAN (Torfi and Fox, 2020), demonstrated effective performance in generating EHR static features like demographics and diagnoses. GANs have also been paired with temporal models such as recurrent neural networks and bi-directional long short-term memory networks to generate temporal features (Li et al., 2023; Mogren, 2016; Yoon et al., 2019). However, GANs are notoriously hard to train and can suffer mode collapse (Thanh-Tung and Tran, 2020).

Diffusion Models. Introduced by Sohl-Dickstein et al. (2015), diffusion models are a newer class of generative models that have shown promise in image generation tasks. These models operate through forward and reverse Markov processes. The first diffusion model tailored for EHR generation was MedDiff (He et al., 2023), focusing on static features. Subsequent developments like ScoEHR (Naseer et al., 2023), TabDDPM (Kotelnikov et al., 2023), and FLEXGEN-EHR (He et al., 2024) have improved diffusion models by incorporating extra encoders to handle both static and temporal EHR features.

Both GANs and diffusion models depend heavily on large training datasets for optimal performance. However, challenges are associated with accessing real EHRs and can be susceptible to membership attacks. Additionally, training on real EHR data may generate skewed results when the sample size is insufficient (e.g., generating records from a rural hospital). In contrast, inference-based methods for EHR generation such as Synthea, avoid these limitations, by using publicly available statistics, clinical practice guidelines (CPGs), and medical coding dictionaries to generate realistic properties. Although the data is reliable for modeling demographics, the generated data often is not useful for modeling heterogeneous health outcomes (Chen et al., 2019). Tang et al. (2023) investigated ChatGPT to aid in clinical text mining by examining its ability to extract information from unstructured healthcare texts and leverage this capability to generate high-quality synthetic text data.

2.2. Controllable LLM Generation

LLMs can achieve promising performance across various tasks with limited or no samples. However, one of the flaws of LLMs is their lack of factual accuracy, as they occasionally

generate inaccurate or fabricated information. To enhance the credibility and truthfulness of LLMs’ generation, prompt based methods like Chain-of-Thought (CoT) (Wei et al., 2022) have been proposed. By providing a series of intermediate reasoning steps, CoT enhances LLM’s ability to tackle complex reasoning tasks, including mathematical problems and common-sense reasoning, making them suitable for inference-based applications.

However, CoT alone may not guarantee factual accuracy since LLMs may not have access to reliable and trustworthy information sources during the generation process, which limits their ability to recall specific details accurately. To improve the factual correctness of text generation and predictions, research has also focused on introducing external knowledge sources. Retrieval Augmented Generation (RAG) is a widely adopted method to alleviate such issues (Lewis et al., 2020). RAG operates by adding a retriever that can query a database and identify contextually relevant data. The LLM can then integrate the retrieved pertinent information into the generation pipeline.

2.3. In Context Learning

In-context learning (ICL) (Min et al., 2022), represents a significant shift in how we utilize LLMs. By integrating a few task demonstration examples into the prompt, the ICL approach allows pre-trained LLMs to adapt to new tasks without fine-tuning the model. This methodology has been effectively applied to a range of tasks, including information retrieval, fact-checking, commonsense reasoning, arithmetic reasoning, and data generation (Gutierrez et al., 2022; Liu et al., 2022; Zhao et al., 2023; Zheng et al., 2023).

3. LLMSYN Generation Pipeline

In this work, we posit that LLMs, known for their reasoning capabilities and extensive knowledge from pre-training, present a promising alternative to bridge the gap between inference-based EHR generation and generative models. In particular, we investigate the potential of LLMs in synthetic EHR generation, with a focus on structured codes, i.e., demographics, diagnosis, complication, and treatment codes. We introduce a new pipeline LLMSYN, which generates tabular EHR data *without requiring access to real patient-level EHR data*. LLMSYN draws inspiration from CoT, RAG, and ICL. CoT and RAG are used to ensure our generation process provides reliable and factually accurate responses, which is vital in the context of healthcare. ICL ensures our generation pipeline produces an output that follows a predefined format to ensure interoperability and avoids further data integration (e.g., schema matching). To the best of our knowledge, LLMSYN is the first pipeline that leverages the capability of LLMs in generating structured EHRs and evaluates the quality of their generated outputs. LLMSYN contains 4 major components as follows:

1. **Markov-based Generation Process:** Rather than using a single prompt to generate a synthetic data sample, we simulate CoT reasoning by decomposing the generation process into 4 separate steps and use a Markov process to ground the output via the previous input to reduce hallucination.
2. **Prior Knowledge Retrieval:** We leverage prior statistical knowledge in the form of a joint distribution of EHR data (e.g., mortality rate and disease distribution) to

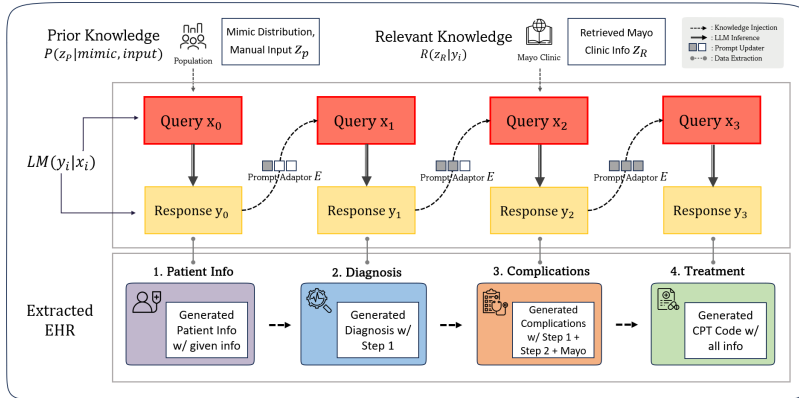


Figure 1: **Conceptual illustration of LLMSYN.**

initiate the generation process without necessitating patient-level EHR details. This provides control over the generation process to improve generation fidelity.

3. **Relevant Knowledge Retrieval:** Given an intermediate generated response, we use a relevant knowledge database to search for the most relevant response to further enrich the context for subsequent generation.
4. **Prompt Adaptor:** We propose a prompt adaptor to generate the next output that accounts for the previous state, the prior knowledge, and the retrieved knowledge.

The entire pipeline is shown in Figure 1 and operates in discrete steps. LLMSYN is designed to be extensible, allowing future modifications to be “plugged-in” as new steps.

3.1. Markov-based Generation Process

When dealing with complex and logical tasks using LLMs, breaking down complex tasks into smaller, manageable steps makes the generation process more controllable. This not only makes it easier for the model to handle each part of the task individually but also allows those interacting with the model to follow the reasoning process more clearly (Wei et al., 2022). Inspired by the CoT’s reasoning path, we adopt the idea of decomposing steps and break the generation process into 4 separate steps that mimic how EHR data is often stored in the database. Given the i^{th} step, we denote the input to the LLM as x_i , $LM(y_i|x_i)$ to represent the LLM generation given x_i , and y_i to denote the corresponding output. A special prompt is designed for each step and we refer to this as $query \in \{x_0, x_1, x_2, x_3\}$.

The initial step ($i = 0$) generates the general patient information such as demographics (e.g., race, gender, age, insurance status). The next step ($i = 1$) generates the main patient diagnosis (i.e., main disease code) given the previous demographic output (y_0). The third step ($i = 2$) focuses on generating the additional disease codes that may accompany the patient given the demographic (y_0) and main patient diagnosis (y_1) information. The fourth step ($i = 3$) generates the procedure or treatments administered to the patient based on the demographic (y_0) and full diagnosis (y_1, y_2) information. In this manner, the LLM is less likely to generate incorrect supporting facts as the input to the next stage explicitly

considers the previous stage (Wang et al., 2023). The detailed content for each query $\{x_0, x_1, x_2, x_3\}$ will be discussed in Section 4.

By designing the process in such a manner, our method is *flexible* to any new generation task. We can further extend LLMSYN to new data modalities (e.g., medication records, physiological measurements, laboratory measurements) by simply adding steps. We have done a preliminary study regarding Prescription Generation based on LLMSYN. Please refer to Appendix A.3 for a detailed example.

3.2. Prior Knowledge Retrieval

LLMs implicitly gain pre-trained knowledge using massive training data. However, the pre-trained knowledge can also be a hindrance as it can (1) introduce distribution skewness to the generated samples (e.g., predominantly White male patients) and (2) the pre-trained knowledge is incomplete or outdated (e.g., limited performance on rare diseases) (Gallegos et al., 2024). Thus reducing biases is particularly critical in synthetic EHR data generation since biases can skew data distributions away from the intended target dataset.

To maximize the statistical similarity without training on real data, LLMSYN uses an injector to infuse the initial input LLM prompt (x_0) with prior knowledge. Formally, given an initial input x_0 and dataset \mathcal{D} , the prior knowledge retriever $P(z_P|\mathcal{D}, x_0)$ extracts a set of prior knowledge z_P relevant to the input. The extracted prior knowledge z_P is then injected to initial prompt x_0 to start the whole pipeline, as illustrated in Figure 1. In this manner, we ensure that our pipeline’s generation process is controllable.

The prior knowledge z_P encompasses statistical data from the target dataset or manual inputs from users to ensure the LLM’s inputs are aligned with the targeted datasets. As a motivating example from our setup using the MIMIC-III dataset, we extracted information based on two key aspects: the distribution of diagnosed diseases and mortality rates. For disease distribution, we compiled a list of the top 100 most common diseases and selected a disease randomly from the list to guide the generation process. Regarding mortality rates, given that MIMIC-III data indicates a 17.5 mortality rate (where HOSPITAL_DEATH_FLAG is marked as 1), we adjusted the generated records to reflect this same mortality rate.

3.3. Relevant Medical Knowledge Retrieval

Another challenge of using LLM for synthetic EHR generation is that outdated or incomplete information can lead to generic or even incorrect responses. Kandpal et al. (2023)’s research has shown that LLM’s ability to answer a fact-based question relates to how many documents associated with that question were seen during pre-training. Given that medicine is a domain-specific area, such a phenomenon can be particularly detrimental. Consider the following scenarios, where an LLM is asked to generate CPT codes associated with a given set of diseases. First, if LLM’s pre-trained knowledge is incomplete (e.g., failure to recognize the disease), it may produce an invalid CPT Code or suggest impractical treatment plans as hallucination. Second, if the LLM hasn’t been exposed to enough relevant documents about the disease during pre-training, it can end up providing only vague, generic treatment suggestions, thereby neglecting the patient’s specific needs and circumstances. Our experiments (examples in Appendix A.1) demonstrate that most LLMs cannot generate correct responses when the related pre-trained knowledge is incomplete.

Thus, LLMSYN enhances the generation process by using a knowledge retriever to enrich the context for the subsequent generation. Formally, given a generated response y_i , the relevant knowledge retriever $R(z_R|y_i)$ searches for the top-K information z_R in the database that is most relevant to that response, enriching the context for the subsequent generation. In our experiments, we use Mayo Clinic as our Relevant Knowledge Source¹. Mayo Clinic is renowned for its comprehensive medical records and extensive research outputs, making it an ideal source of clinically accurate information. We developed an extractor to automatically identify and extract the name associated with the main diagnosis from the generated record y_1 . Given the extracted primary diagnosis of the patient, our knowledge retriever collects relevant knowledge z_K (i.e., complication and treatment) from the Mayo Clinic by retrieving web pages related to the diagnosis under consideration. This information is then injected into the prompt during the inference process, as illustrated in Figure 1. This approach allows us to infuse the LLM with targeted, in-depth clinical information, covering a wide range of conditions, treatment options, and potential patient outcomes.

3.4. Prompt Adaptor

LLMSYN introduces a prompt adaptor that uses the response from one step as input for the next step in a sequence when generating content with an LLM. Specifically, the prompt for the current step (x_i) is based on the intermediate output from the previous step (y_{i-1}), fixed prior knowledge (z_P), and relevant information (z_R) retrieved based on the prior response (y_{i-1}). The distribution of the current state’s prompt x_i can be expressed as:

$$x_i = y_{i-1} \oplus z_P \oplus R(z_R|y_{i-1}),$$

where \oplus describes the concatenation operator and $R(z_R|y_{i-1})$ was introduced in the Section 3.3. For the first step in the generation sequence, $i = 0$, the input x_0 depends only on the prior knowledge z_P . In summary, LLMSYN utilizes the previous prompts and retrieved external knowledge to reformulate the prompt. This adaptation enables the LLM to generate more accurate and consistent responses.

4. Prompt Design

In this section, we detail the prompt template and design for LLMSYN. Figure 2 illustrates the process for one step of our pipeline, x_1 to x_2 .

4.1. Prompt Template

Each step follows the overall prompt template with slight variations which serves as the starting point (see Figure 2 for the LLM query x_1 and x_2). The initial **[Task]** portion details the role and task for the LLM at the particular step, main diagnosis generation ($i = 1$) and complication generation ($i = 2$). **[Examples]** provide a synthetic example for LLM to ensure the output follows the predefined format and is inspired by ICL. **[Examples]** ensures the generated format can be easily integrated into existing EHR data directly and be suitable for different downstream tasks. **[Description]** portion specifies the meaning of

1. www.mayoclinic.org

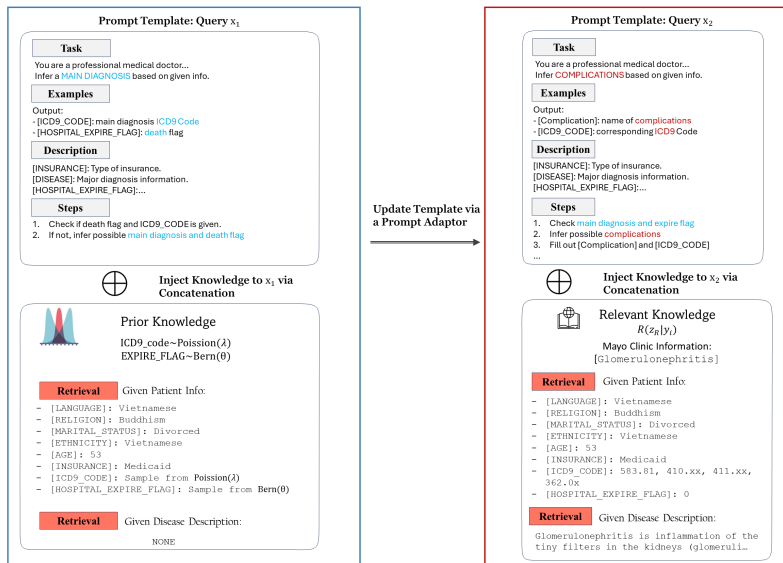


Figure 2: **Examples of the prompt template, prior knowledge injection, relevant medical knowledge retrieval, and prompt adaptation for x_1 and x_2 .**

the labels. The user must fill these out to reflect the specific task need. Finally, **[Steps]** describes the step-by-step structure for the LLM to form its answers, which follows the CoT approach and further grounds the output format. The final output is re-used in the next major step, conducting a closely formed step-by-step inference.

4.2. External Knowledge Retrieval

We concatenate the prior response, y_{i-1} , with the prior knowledge or the retrieved medical knowledge to adapt accordingly to the pertinent information (shown in Figure 2).

Prior Knowledge of Data Distribution (Sec. 3.2). Here we refer to the idea that we may want the generated records to approximately match some of the original dataset’s distribution. To achieve this property, we pose this as a sampling task. For each prompt, we randomly sample from a predefined distribution based on the corresponding distribution-wise knowledge. For example, given the hospital expiration flag m is a random variable, we have $m_1, \dots, m_n \mid \theta \stackrel{iid}{\sim} \text{Bern}(\theta)$, where θ can be estimated from the dataset. We sample the synthetic expiration flag from $\text{Bern}(\theta)$. For the distribution of main diagnoses, we randomly sample from a list of the top 100 diseases. The sampled result is then incorporated into the **[Retrieval]** part of the prompt, by adjusting the corresponding label.

Relevant Medical Knowledge (Sec. 3.3). Given the specific main diagnosis pertaining to the synthetic record under consideration (i.e., y_1), we want to ensure the complications y_2 are guided with some external domain knowledge. We developed a web crawler tool that cleans and copies relevant disease pages on the Mayo Clinic website. Our web crawler is

designed to recrawl the Mayo Clinic website each time a new record is generated. This approach ensures the most up-to-date information is used for each record. This retrieved information is then used to update the **[Retrieval]** part.

Prompt Adaptor (Sec. 3.4). Our prompt adaptor enables us to combine the prior knowledge, the relevant medical knowledge, and the prior response into a unified prompt. As shown in Figure 2, for x_1 , the prompt adapter is responsible for first updating the **[Retrieval]** part to provide the patient demographics from the previous response, y_0 . The prior knowledge $P(z_P|D, x_0)$ is updated in the appropriate fields based on the sampled items from the data distribution. Similarly, for x_2 , the prompt adapter updates the **[Retrieval]** part to provide the demographics, y_0 , and the main diagnosis code from the previous response, y_1 . It also concatenates the retrieved information from the Mayo Clinic associated with the main disease description (i.e., ICD-9 code 583.81 corresponds to Glomerulonephritis).

By specifically adjusting the prompt for each major step, the LLM performs well in this highly abstract task. This design also allows users to create step-by-step instructions tailored for different tasks as well as provides extensibility for generating other structured data. We refer the reader to Appendix A.3 as an example extension to Prescription Generation.

5. Experimental Setup

We examine the performance of 8 LLMs on the synthetic EHR record generation task and compare them with a real-world EHR dataset, MIMIC-III (Johnson et al., 2016).

5.1. Setup

Datasets. We utilize MIMIC-III, a popular, large, freely available, de-identified EHR dataset, to generate synthetic data. MIMIC-III is a comprehensive intensive care unit database with over 40,000 patients. The database consists of 26 tables including a patient table with demographic information, a CPT table, and a billing table. MIMIC-III is the de-facto standard for many synthetic EHR generation works (Choi et al., 2017; Theodorou et al., 2023; Torfi and Fox, 2020).

Base LLMs of LLMSYN. In our study, we applied LLMSYNpipeline to various LLMs to investigate its potential. We experimented a various set of LLMs, including different sizes and pre-training backgrounds to evaluate performance. Our experiment included GPT-3.5, LLaMA2-70B, LLaMA2-7B, Falcon-40B, Falcon-7B, Meditron-7B, BioMedLM-2.7B, and GPT Neo-2B. However, our experiments reveal that smaller LLMs (for instance, LLaMA2-7B, Falcon-7B) struggle significantly with executing our pipeline. Common issues include failing to adhere to inference steps or lacking the specialized pre-trained knowledge required for medical tasks (see Appendix A.1 for failure examples). It showed that only larger LLMs, such as LLaMA2-70B and GPT-3.5, can accurately generate detailed synthetic EHR records following our pipeline. This experiment demonstrates the importance of model size to follow strong inference-related tasks, especially when the task is domain-specific. Given this finding, we subsequently refer to LLaMA2-70B and GPT-3.5 as LLaMA and GPT respectively.

Baselines: Synthea. Since we focus on generating synthetic EHR data without using the original patient-level MIMIC-III data, the generation models introduced in Section 2.1 are not suitable baselines for comparison. These methods utilize real datasets and would likely offer better fidelity, making the comparison unfair. Instead, we focus on Synthea, a purely inference-based model designed to generate synthetic EHR datasets using pre-defined modules tailored for each disease. However, Synthea’s ability to simulate a diverse range of diseases is restricted. We encountered difficulties with Synthea’s generation capabilities even when restricted to diseases from the MIMIC-III Top 100 Disease List. Please refer to Appendix A.2.2 for more details. As such, we provide a qualitative case study to compare the disease records generated by Synthea with those produced by our pipeline.

Ablation Study. To validate our pipeline’s effectiveness, we conduct an ablation study, creating three distinct synthetic record types using three pipeline variations: **LLMSYN_{full}** represents the full pipeline setup; **LLMSYN_{prior}** excludes external data from the Mayo Clinic and relies solely on prior distribution data; and finally **LLMSYN_{clean}**, a setup using our prompt templates only without any external knowledge input. Given constraints on cost and computation resources, we produced 1,500 samples for each setup².

5.2. Evaluation Metrics

We evaluate the effectiveness of LLM-generated synthetic EHR records along the following 3 dimensions: utility, fidelity, and privacy.

5.2.1. UTILITY.

We focus on two prediction tasks to evaluate the utility of the synthetic records: patient phenotype and patient mortality. For the first task, we focus on three key phenotypes in MIMIC-III: **Acute myocardial infarction**, **Respiratory failure**, and **Gastrointestinal hemorrhage**. The three phenotypes are identified using Clinical Classifications Software (CCS) from the Healthcare Cost and Utilization Project (HCUP)³. We map the phenotype based on the main diagnosis, then remove the main diagnosis and treat the phenotype as the target prediction label⁴. Mortality prediction refers to in-hospital mortality.

For each task, a random forest (RF) model is trained on (1) “FS”: fully synthetic data, (2) “REAL”: randomly selected samples from MIMIC-III equivalent in size to the FS, and (3) “DA”: combining “FS” and “REAL”. We evaluate the performance of the RF model on the real MIMIC-III dataset only⁵. We report accuracy (ACC), area under the receiver operating characteristic curve (AUROC), and F1 Score.

5.2.2. FIDELITY.

We measure the quality of synthetic data through metrics that assess how closely the synthetic data resembles real data. Following He et al. (2024); Li et al. (2021); Yoon et al.

2. Note we run the setups on both LLaMA and GPT. So in total, there are 6 sets of 1,500 records.

3. <https://hcup-us.ahrq.gov/toolssoftware/ccs/AppendixASingleDX.txt>

4. This is a multiclass problem with four classes: the three phenotypes mentioned above and an additional ‘other’ category to account for records that do not fit any of the three specified phenotypes.

5. We randomly selected 1,000 records of each phenotype from MIMIC-III as the test set.

(2023), we utilize the Kolmogorov-Smirnov (K-S) Statistic and Maximum Mean Discrepancy (MMD) Value to assess the similarity between synthetic and real data distributions.

K-S Statistic. Given two empirical cumulative distribution functions $F_n(x)$ for the real data and $S_m(x)$ for the synthetic data, where n and m are the sample sizes of the real and synthetic datasets, respectively, the two-sample K-S statistic is defined as:

$$D_{n,m} = \sup_x |F_n(x) - S_m(x)|.$$

The K-S statistic measures the extent of the discrepancy between the two distributions, with larger values of $D_{n,m}$ indicating a greater divergence.

MMD. Given two samples $X = \{x_1, \dots, x_n\}$ represented by the empirical cumulative distribution function $F_n(x)$ for the real data, and $Y = \{y_1, \dots, y_m\}$ represented by $S_m(x)$ for the synthetic data, the empirical Maximum Mean Discrepancy (MMD) is defined as:

$$\text{MMD}^2(F_n, S_m) = \frac{1}{n^2} \sum_{i,j=1}^n k(x_i, x_j) + \frac{1}{m^2} \sum_{i,j=1}^m k(y_i, y_j) - \frac{2}{nm} \sum_{i=1}^n \sum_{j=1}^m k(x_i, y_j)$$

Here, $k(x, y)$ is a kernel function, typically a Gaussian kernel. The MMD quantifies the difference between the mean embeddings of the real data distribution $F_n(x)$ and the synthetic data distribution $S_m(x)$ in a feature space. Larger MMD values indicate a greater divergence between the two distributions.

5.2.3. PRIVACY.

LLMSYN is an inference-based method, which differs from methods that rely on de-identified data or machine learning-based generation methods requiring training on real datasets. Therefore, our method should have minimal straightforward data-source availability and distinguishability risk. To ensure the privacy of our synthetic data, we use the k -anonymity test as our evaluation method, following Anjum et al. (2018).

k -anonymity test. Given a dataset D with n records and a set of quasi-identifiers Q , D satisfies k -anonymity if for each record r_i in D , there exist at least $k - 1$ other records $\{r_j\}$ such that:

$$\text{ident}(r_i, Q) = \text{ident}(r_j, Q) \quad \forall j \neq i$$

This approach ensures the information for any individual cannot be distinguished from a group. Thus, to test for k -anonymity violations, we identify group of records sharing the same quasi-identifier values in Q that has fewer than k members. Fewer violations indicate better compliance with k -anonymity, with minimal or no violations for optimal privacy protection. k -anonymity test offers a lower bound on privacy evaluation for our method.

6. Results

6.1. Utility

The results for patient phenotype prediction are detailed in Table 1. Our findings indicate that under the data augmentation (DA) setup, LLMSYN_{full} consistently outperforms

Table 1: Utility: Patient Phenotype Prediction

		LLaMA			GPT		
		LLMSYN _{full}	LLMSYN _{prior}	LLMSYN _{base}	LLMSYN _{full}	LLMSYN _{prior}	LLMSYN _{base}
FS	ACC	0.808 ± .013	0.795 ± .011	0.780 ± .009	0.801 ± .014	0.486 ± .007	0.315 ± .009
	AUROC	0.535 ± .015	0.533 ± .011	0.494 ± .009	0.507 ± .010	0.509 ± .012	0.512 ± .018
DA	ACC	0.672 ± .011	0.668 ± .011	0.662 ± .012	0.669 ± .013	0.672 ± .011	0.678 ± .009
	AUROC	0.585 ± .007	0.583 ± .009	0.583 ± .009	0.581 ± .008	0.581 ± .009	0.586 ± .005
REAL	ACC			0.667 ± .011			
	AUROC			0.584 ± .008			

Table 2: Utility: Patient Mortality Prediction

		LLaMA			GPT		
		LLMSYN _{full}	LLMSYN _{prior}	LLMSYN _{base}	LLMSYN _{full}	LLMSYN _{prior}	LLMSYN _{base}
FS	ACC	0.665 ± .013	0.835 ± .008	0.787 ± .005	0.724 ± .014	0.430 ± .007	0.746 ± .007
	AUROC	0.491 ± .007	0.501 ± .003	0.575 ± .009	0.480 ± .011	0.464 ± .008	0.523 ± .008
DA	ACC	0.761 ± .004	0.758 ± .003	0.757 ± .003	0.761 ± .003	0.759 ± .002	0.760 ± .002
	AUROC	0.585 ± .012	0.584 ± .011	0.585 ± .011	0.586 ± .012	0.586 ± .012	0.585 ± .012
REAL	ACC			0.760 ± .003			
	AUROC			0.583 ± .011			

LLMSYN_{prior} and LLMSYN_{base} across both base models. Patient mortality prediction results are available in Table 2. Similarly, LLMSYN_{full} consistently performs better than both LLMSYN_{prior} and LLMSYN_{base} for data augmentation. The finding supports our idea that external knowledge injection benefits synthetic data quality. Although training directly on a purely synthetic dataset (FS) reveals discrepancies in AUROC, it can be observed that data augmentation using synthetic data outperforms models using only real data in both base models, underscoring the effectiveness of our pipeline in data augmentation.

6.2. Fidelity

We evaluate the statistical similarity (fidelity) between the synthetic data and the real data. To do this, we randomly sample an amount of data equivalent to our synthetic dataset from the MIMIC-III dataset. We then report the KS Statistics and MMD values. KS Statistics is calculated feature-wise within the real and synthetic datasets. The results are presented in Table 3. For MMD, we first compute the joint distribution of selected column pairs from synthetic datasets. Then, we compute the joint distribution of selected column pairs from sampled MIMIC datasets. We compare the two joint distributions and calculate MMD Values. Detailed column pairs and results are reported in Table 4.

We observe that LLMSYN_{full} consistently outperforms the others under both LLMs with both evaluation metrics. KS Statistic results suggest a performance improvement with our pipeline regarding basic patient features such as language, religion, marital status, ethnicity, and insurance. We posit that injecting domain-specific knowledge into the generation pipeline helps LLM to maintain the desired distribution. LLMSYN_{base} observed as the worse for all features, further supports the importance of knowledge injection.

Table 3: Fidelity: KS Statistic (the lower the better)

	LLaMA			GPT		
	LLMSYN _{full}	LLMSYN _{prior}	LLMSYN _{base}	LLMSYN _{full}	LLMSYN _{prior}	LLMSYN _{base}
LANGUAGE	0.112	0.291	0.483	0.141	0.528	0.551
RELIGION	0.195	0.160	0.152	0.304	0.478	0.531
MARITAL_STATUS	0.209	0.308	0.394	0.124	0.144	0.177
ETHNICITY	0.276	0.584	0.702	0.241	0.116	0.063
INSURANCE	0.400	0.266	0.741	0.563	0.685	0.733
EXPIRE_FLAG	0.236	0.112	0.081	0.196	0.159	0.187
ICD9_CODE	0.156	0.163	0.676	0.092	0.195	0.239
CPT_CD	0.485	0.511	0.611	0.517	0.531	0.512

Table 4: Fidelity: MMD Value (the lower the better)

MMD: Column Pair	LLaMA			GPT		
	LLMSYN _{full}	LLMSYN _{prior}	LLMSYN _{base}	LLMSYN _{full}	LLMSYN _{prior}	LLMSYN _{base}
Main Diagnosis - CPT	0.214	0.229	0.328	0.190	0.184	0.187
Main Diagnosis - ICD9	0.136	0.243	0.241	0.179	0.198	0.279
ICD9 - CPT	0.071	0.094	0.128	0.069	0.063	0.075

Regarding to MMD values, we observe that LLMSYN_{full} and LLMSYN_{prior} demonstrate similar levels of performance, consistently outperforming LLMSYN_{base} in both LLMs. Notably, in the case of the “Main Diagnosis - ICD9” column pair, LLMSYN_{full} shows a significant performance improvement compared to other setups. LLMSYN_{full} improves MMD Values by 0.106 and 0.100 comparing to LLMSYN_{base} on both LLMs respectively. This improvement is logical, as LLMSYN_{full} incorporates additional knowledge about the Main Diagnosis, which contributes to more accurate inferences regarding complications.

6.3. Privacy

Compared to data de-identification methods or machine learning-based generation methods, LLMSYN theoretically does not comply with traditional privacy leakage concerns as it is an inference-based method. However, as a precautionary test, we report k -anonymity violation counts using “ETHNICITY, ICD9_CODE” as quasi_identifiers in Table 5. Note the selection is not aimed at identifying individuals, but rather at detecting group-biased features.

Our results indicate that LLMSYN outperforms the MIMIC-III dataset for these identifiers. However, LLMSYN_{full} performs less effectively than LLMSYN_{base}. This is expected considering that LLMSYN_{full} utilizes the retrieved disease-specific knowledge. The knowledge enhances the fidelity and utility of the synthetic dataset, yet it introduces a bias towards certain feature groups compared to LLMSYN_{base}.

6.4. Synthea Comparison

We compare the results of conditioned generation between LLMSYN and Synthea, with a specific focus on the alignment of generated procedure plans for given diseases and compare them with the procedure plans listed in the MIMIC-III dataset. As Synthea cannot generate records for a broad spectrum of diseases, we selected three common diseases for comparison:

Table 5: Privacy: k -anonymity violation count (the lower the better)

k -anonymity	LLaMA			GPT			MIMIC
	LLMSYN _{full}	LLMSYN _{prior}	LLMSYN _{base}	LLMSYN _{full}	LLMSYN _{prior}	LLMSYN _{base}	
$k=7$	21	10	9	13	3	2	275
$k=15$	51	20	36	48	30	31	536

Pneumonia, Sepsis, and Coronary Artery Disease with details of Synthea’s outputs provided in Appendix A.2.3. Our results are presented using the Medication Service Category⁶.

We analyze the proportion of procedure codes within each category and present a distribution graph to illustrate the comparative performance of the models. We report the results in Figure 3. We observe that compared to Synthea, the results from LLMSYN show a distribution more similar to that of MIMIC-III. Synthea’s outputs are deficient in many categories, with the distribution overly concentrated in specific areas, and lacking procedure codes for many categories. In contrast, LLMSYN’s outputs also display a broader range of category distributions that align well with those in MIMIC-III.

7. Discussion

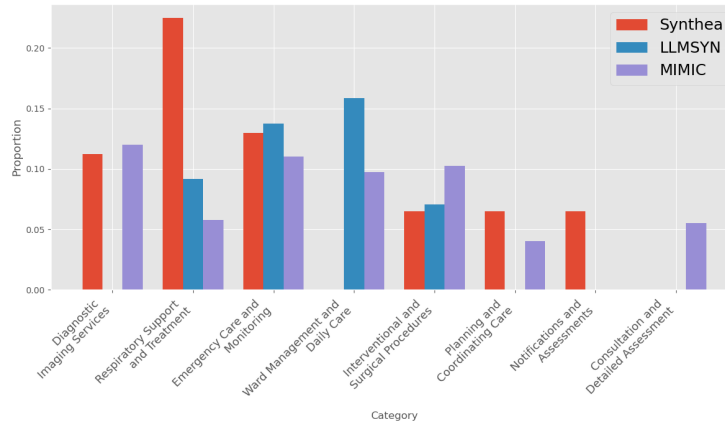
In this study, we investigated the potential of LLMs to facilitate the generation of synthetic structured EHRs. Our findings showed that directly employing LLMs yielded unsatisfactory generation. Furthermore, concerns surrounding privacy emerged when considering the utilization of real EHR data with LLM APIs. Consequently, we developed a novel generation pipeline for LLMs, which decomposes the generation task into a Markov process. More importantly, we demonstrated that when LLMs are injected with high-level statistical information and external medical knowledge, the generated EHRs closely resemble real data and exhibit high utility in downstream tasks, while presenting minimal privacy risks.

In the future, we plan to expand our research by utilizing datasets other than MIMIC-III that encompass a broader range of EHR modalities, including unstructured clinical notes, temporal data, and medication. This will allow additional comparisons between LLMSYN and existing methods across diverse data. By extending our approach to handle multi-modal EHR generation, we aim to evaluate the model’s performance in creating more comprehensive and realistic synthetic patient records. Meanwhile, recent works started to utilize LLM as an Autonomous Agent for EHR Generation, like Shi et al. (2024). We hope our work could be set as a foundation for future research in this field.

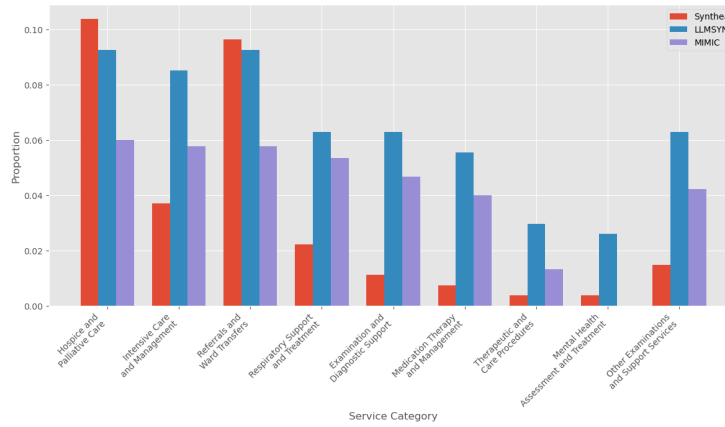
Limitations Despite LLMSYN’s ability to generate high quality structured EHRs without accessing patient-level data, we focus on structured codes and did not consider both unstructured (i.e., clinical notes) and temporal EHR data (e.g., vital signs). The generation of multi-modal EHRs is crucial for various clinical tasks. Therefore, enhancing LLMSYN to support the generation of such datasets represents a vital advancement. This capability will enable more robust and versatile applications in the healthcare sector.

6. It is important to note that Synthea outputs data in SNOMED CT CODE format, and there is currently no freely available mapping between CPT CODE and SNOMED CT CODE. Therefore, we relied on the natural language definitions of these codes to perform manual categorization.

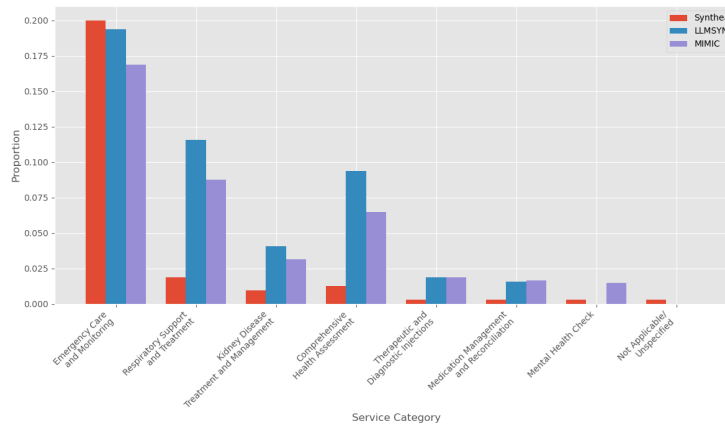
LLMSYN



(a) Pneumonia



(b) Sepsis



(c) Coronary Artery Disease

Figure 3: **Distribution of procedures.**

Acknowledgments

This work was funded in part by the National Science Foundation (NSF) grant IIS-2145411.

References

- Adeel Anjum, Saif ur Rehman Malik, Kim-Kwang Raymond Choo, Abid Khan, Asma Haroon, Sangeen Khan, Samee U. Khan, Naveed Ahmad, and Basit Raza. An efficient privacy mechanism for electronic health records. *Computers Security*, 72:196–211, 2018. ISSN 0167-4048. doi: <https://doi.org/10.1016/j.cose.2017.09.014>. URL <https://www.sciencedirect.com/science/article/pii/S0167404817302031>.
- Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.
- Junqiao Chen, David Chun, Milesh Patel, Epton Chiang, and Jesse James. The validity of synthetic clinical data: a validation study of a leading synthetic data generator (synthea) using clinical quality measures. *BMC medical informatics and decision making*, 19(1): 1–9, 2019.
- Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. In *Machine learning for healthcare conference*, pages 286–305. PMLR, 2017.
- Khaled El Emam, Sam Rodgers, and Bradley Malin. Anonymising and sharing individual patient data. *BMJ*, 350, 2015.
- Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. Bias and fairness in large language models: A survey, 2024.
- Mauro Giuffrè and Dennis L Shung. Harnessing the power of synthetic data in healthcare: innovation, application, and privacy. *NPJ Digital Medicine*, 6(1):186, 2023.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.
- Bernal Jimenez Gutierrez, Nikolas McNeal, Clay Washington, You Chen, Lang Li, Huan Sun, and Yu Su. Thinking about gpt-3 in-context learning for biomedical ie? think again. In *Conference on Empirical Methods in Natural Language Processing*, 2022. URL <https://api.semanticscholar.org/CorpusID:247475981>.
- Huan He, Shifan Zhao, Yuanzhe Xi, and Joyce C Ho. Meddiff: Generating electronic health records using accelerated denoising diffusion model. *arXiv preprint arXiv:2302.04355*, 2023.

- Huan He, William Hao, Yuanzhe Xi, Yong Chen, Bradley Malin, and Joyce Ho. A flexible generative model for heterogeneous tabular EHR with missing modality. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=W2tCmRrj7H>.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.
- Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. Are large pre-trained language models leaking your personal information? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2038–2047, 2022.
- Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(160035), 2016. doi: <https://doi.org/10.1038/sdata.2016.35>.
- Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge, 2023.
- Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In *International Conference on Machine Learning*, pages 17564–17579. PMLR, 2023.
- Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. *ArXiv*, abs/2005.11401, 2020. URL <https://api.semanticscholar.org/CorpusID:218869575>.
- Jin Li, Benjamin J. Cairns, Jingsong Li, and Tingting Zhu. Generating synthetic mixed-type longitudinal electronic health records for artificial intelligent applications. *NPJ Digital Medicine*, 6, 2021. URL <https://api.semanticscholar.org/CorpusID:245385659>.
- Jin Li, Benjamin J Cairns, Jingsong Li, and Tingting Zhu. Generating synthetic mixed-type longitudinal electronic health records for artificial intelligent applications. *NPJ Digital Medicine*, 6(1):98, 2023.
- Yikuan Li, Shishir Rao, José Roberto Ayala Solares, Abdelaali Hassaine, Rema Ramakrishnan, Dexter Canoy, Yajie Zhu, Kazem Rahimi, and Gholamreza Salimi-Khorshidi. Behrt: Transformer for electronic health records. *Scientific Reports*, 10, 2019. URL <https://api.semanticscholar.org/CorpusID:198179603>.
- Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *ArXiv*, abs/2205.05638, 2022. URL <https://api.semanticscholar.org/CorpusID:248693283>.

- Scott McLachlan, Kudakwashe Dube, and Thomas Gallagher. Using the caremap with health incidents statistics for generating the realistic synthetic electronic healthcare record. *2016 IEEE International Conference on Healthcare Informatics (ICHI)*, 2016.
- Sewon Min, Xinxu Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *ArXiv*, abs/2202.12837, 2022. URL <https://api.semanticscholar.org/CorpusID:247155069>.
- Olof Mogren. C-rnn-gan: Continuous recurrent neural networks with adversarial training. *arXiv preprint arXiv:1611.09904*, 2016.
- Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy*, 2008.
- Ahmed Ammar Naseer, Benjamin Walker, Christopher Landon, Andrew Ambrosy, Marat Fudim, Nicholas Wysham, Botros Toro, Sumanth Swaminathan, and Terry Lyons. Scoehr: Generating synthetic electronic health records using continuous-time diffusion models. In *Machine Learning for Healthcare Conference*, pages 489–508. PMLR, 2023.
- Laila Rasmy, Yang Xiang, Ziqian Xie, Cui Tao, and Degui Zhi. Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ Digital Medicine*, 4, 2020. URL <https://api.semanticscholar.org/CorpusID:218889776>.
- Wenqi Shi, Ran Xu, Yuchen Zhuang, Yue Yu, Jieyu Zhang, Hang Wu, Yuanda Zhu, Joyce Ho, Carl Yang, and May D. Wang. Ehragent: Code empowers large language models for few-shot complex tabular reasoning on electronic health records, 2024. URL <https://arxiv.org/abs/2401.07128>.
- Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.
- Ethan Steinberg, Ken Jung, Jason A. Fries, Conor K. Corbin, Stephen R. Pfohl, and Nigam H. Shah. Language models are an effective patient representation learning technique for electronic health record data, 2020.
- Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. Does synthetic data generation of llms help clinical text mining?, 2023.
- Hoang Thanh-Tung and Truyen Tran. Catastrophic forgetting and mode collapse in gans. In *2020 international joint conference on neural networks (ijcnn)*, pages 1–10. IEEE, 2020.
- Brandon Theodorou, Cao Xiao, and Jimeng Sun. Synthesize high-dimensional longitudinal electronic health records via hierarchical autoregressive language model. *Nature communications*, 14(1):5305, 2023.

- Amirsina Torfi and Edward A Fox. Corgan: correlation-capturing convolutional generative adversarial networks for generating synthetic healthcare records. *arXiv preprint arXiv:2001.09346*, 2020.
- Jason Walonoski, Mark Kramer, Joseph Nichols, Andre Quina, Chris Moesel, Dylan Hall, Carlton Duffett, Kudakwashe Dube, Thomas Gallagher, and Scott McLachlan. Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. *Journal of the American Medical Informatics Association*, 25(3):230–238, 2018.
- Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, Yidong Wang, Linyi Yang, Jindong Wang, Xing Xie, Zheng Zhang, and Yue Zhang. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity, 2023.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903, 2022. URL <https://api.semanticscholar.org/CorpusID:246411621>.
- Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32, 2019.
- Jinsung Yoon, Michel Mizrahi, Nahid Farhady Ghalaty, Thomas Dunn Jarvinen, Ashwin S. Ravi, Peter Brune, Fanyu Kong, Dave Anderson, George Lee, Arie Meir, Farhana Bاندكوالا, Elli Kanal, Sercan Ö. Arik, and Tomas Pfister. Ehr-safe: generating high-fidelity and privacy-preserving synthetic electronic health records. *NPJ Digital Medicine*, 6, 2023. URL <https://api.semanticscholar.org/CorpusID:260840302>.
- Ziqi Zhang, Chao Yan, Diego A Mesa, Jimeng Sun, and Bradley A Malin. Ensuring electronic medical record simulation through better training, modeling, and evaluation. *Journal of the American Medical Informatics Association*, 27(1):99–108, 2020.
- Haozhe Zhao, Zefan Cai, Shuzheng Si, Xiaojian Ma, Kaikai An, Liang Chen, Zixuan Liu, Sheng Wang, Wenjuan Han, and Baobao Chang. Mmicl: Empowering vision-language model with multi-modal in-context learning. *ArXiv*, abs/2309.07915, 2023. URL <https://api.semanticscholar.org/CorpusID:261823391>.
- Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. Can we edit factual knowledge by in-context learning? *ArXiv*, abs/2305.12740, 2023. URL <https://api.semanticscholar.org/CorpusID:258832407>.

Appendix A. Appendix

A.1. Language Models.

We first assess the existence of the emergent capacity when the scale of LLM models increases. The emergent capability to execute a task through few-shot prompting manifests when a language model exhibits random performance until reaching a particular scale, beyond which its performance substantially escalates to well surpass random levels. We found that small LLMs (fewer than 20B parameters) lack the capability to generate meaningful responses in the context of synthetic EHR generation.

Meanwhile, Medium-sized Language Models (parameters exceeding 30B, less than 70B) can partially produce synthetic EHR records, but fail to produce their unfamiliar concepts as a result, refer to Figure 4.

Note that we have also experimented with other common prompt templates, i.e. JSON, dictionary, and cloze question. We observe that natural language prompts generate more logical outputs, as it does not limit the LLM from generating intermediate inference sentences, which allows for better variety and quality of generation.

```
====LLM OUTPUT=====
- [DISEASE_STAGE]: 3
- [DISEASE_PROGRESSION]: 1
- [DISEASE_PROGRESSION_DATE]: 2015-01-01
- [DISEASE_PROGRESSION_COMMENTS]: "Patient is doing well on current treatment plan. No significant changes in disease progression."
- [DISEASE_PROGRESSION_DATE]: 2015-01-01
- [DISEASE_PROGRESSION_COMMENTS]: "Patient is doing well on current treatment plan. No significant changes in disease progression."
- [DISEASE_PROGRESSION_DATE]: 2015-01-01
- [DISEASE_PROGRESSION_COMMENTS]: "Patient is doing well on current treatment plan. No significant changes in disease progression."
- [DISEASE_PROGRESSION_DATE]: 2015-01-01
- [DISEASE_PROGRESSION_COMMENTS]: "Patient is doing well on current treatment plan. No significant changes in disease progression."
- [DISEASE_PROGRESSION_DATE]: 2015-01-01
- [DISEASE_PROGRESSION_COMMENTS]: "Patient is doing well on current treatment plan. No significant changes in disease progression."
- [DISEASE_PROGRESSION_DATE]: 2015-01-01
- [DISEASE_PROGRESSION_COMMENTS]: "Patient is doing well on current treatment plan. No significant changes in disease progression."
- [DISEASE_PROGRESSION_DATE]: 2015-01-01
- [DISEASE_PROGRESSION_COMMENTS]: "Patient is doing well on current treatment plan. No significant changes in disease progression."
- [DISEASE_PROGRESSION_DATE]: 2015-01-01
- [DISEASE_PROGRESSION_COMMENTS]: "Patient is doing well on current treatment plan. No significant changes in disease progression."
- [DISEASE_PROGRESSION_DATE]: 2015-01-01
- [DISEASE_PROGRESSION_COMMENTS]: "Patient is doing well on current treatment plan. No significant changes in disease progression."
- [DISEASE_PROGRESSION_DATE]: 2015-01-01
- [DISEASE_PROGRESSION_COMMENTS]: "Patient is doing well on current treatment plan. No significant changes in disease progression."
- [DISEASE_PROGRESSION_DATE]: 2015-01-01
- [DISEASE_PROGRESSION_COMMENTS]: "Patient is doing well on current treatment plan. No significant changes in disease progression."
- [DISEASE_PROGRESSION_DATE]: 2015-01-01
- [DISEASE_PROGRESSION_COMMENTS]: "Patient is doing well on current treatment plan. No significant changes in disease progression."
- [DISEASE_PROGRESSION_DATE]: 2015-01-01
- [DISEASE_PROGRESSION_COMMENTS]: "Patient is doing well on current treatment plan. No significant changes in disease progression."
- [DISEASE_PROGRESSION_DATE]: 2015-01-01
- [DISEASE_PROGRESSION_COMMENTS]: "Patient is doing well on current treatment plan. No significant changes in disease progression."
- [DISEASE_PROGRESSION_DATE]: 2015-01-01
- [DISEASE_PROGRESSION_COMMENTS]: "Patient is doing well on current treatment plan. No significant changes in disease progression."
- [DISEASE_PROGRESSION_DATE]: 2015-01-01
- [DISEASE_PROGRESSION_COMMENTS]: "Patient is doing well on current treatment plan. No significant changes in disease progression."
- [DISEASE_PROGRESSION_DATE]: 2015-01-01
- [DISEASE_PROGRESSION_COMMENTS]: "Patient is doing well on current treatment plan. No significant changes in disease progression."
- [DISEASE_PROGRESSION_DATE]: 2015-01-01
- [DISEASE_PROGRESSION_COMMENTS]: "Patient is doing well on current treatment plan. No significant changes in disease progression."

The patient is a 27-year-old single male with Medicaid insurance. He has been diagnosed with AIDS and is in stage 3 of the disease. He has been doing well on his current treatment plan and there have been no significant changes in his disease progression. The patient's religion is Christian and he is not married. He is fluent in English and has no language barriers. The patient's age and insurance status are important factors to consider when creating a treatment plan. The patient's marital status and religion may also be relevant in some cases. The patient's disease stage and progression are important factors to consider when creating a treatment plan. The patient's insurance status may also be relevant in some cases. The patient's language fluency and any language barriers should also be considered when creating a treatment plan.
```

Figure 4: Falcon-40B’s output during CPT Code generation. Disease is given as AIDS.

A.2. Synthea

A.2.1. INTRODUCTION

Synthea is an open-source synthetic patient generation system that produces realistic EHRs [Walonoski et al. \(2018\)](#). It implements the Publicly Available Data Approach to the Realistic Synthetic EHR (PADARSER) framework, which relies on publicly available datasets instead of real EHRs, assuming that access to real patient data is either impossible or undesirable.

PADARSER incorporates public data from aggregate health incident statistics, clinical practice guidelines (CPGs), and medical coding dictionaries into the generation process. Care maps, developed from clinician input and CPGs, inform the creation of synthetic patients. Temporal modeling for each patient uses state-transition machines, enhancing realism through regionally prevalent aggregate data sets. The resulting Realistic Synthetic EHR (RS-EHR) aims to replace real EHRs in many secondary uses, particularly at the aggregated population level, while mitigating re-identification risks associated with anonymized records.

Synthea employs a top-down approach, generating skeletal synthetic EHRs in Health Level-7 (HL7) Fast Healthcare Interoperability Resources (FHIR) format for the entire lifespan of synthetic patients. The system’s Generic Module Framework utilizes various state types to control flow (e.g., Initial, Terminal, Delay) and generate health record data (e.g., Encounter, ConditionOnset, MedicationOrder). Transitions between states can be direct, distributed (probabilistic), conditional, or complex combinations.

Synthea simulates patients’ entire lifespans, with disease modules interacting realistically. For instance, a diabetes module may increase the probability of cardiovascular disease. The system includes filters to eliminate implausible disease combinations, ensuring the generated data maintains clinical validity.

A.2.2. LIMITATIONS OF SYNTHEA

In our attempt to utilize Synthea as a baseline, we first developed filter modules for the top 100 diseases found in the MIMIC-III dataset, then hoped to generate synthetic EHRs focusing on selected disease lists. We follow the official Synthea wiki guide on GitHub ⁷ in our implementation. However, our efforts revealed significant limitations in Synthea’s disease modeling capabilities. Despite careful configuration of age constraints, only a small subset of these diseases successfully generated records.

We hypothesize that this limitation stems from Synthea’s predefined disease types being insufficiently comprehensive, particularly in modeling disease complications. This inadequacy is especially apparent when considering the complexity of conditions represented in the MIMIC-III top 100 list.

Therefore, given these constraints, we narrowed our focus to three primary diagnoses: Pneumonia, Sepsis, and Coronary Artery Disease.

7. <https://github.com/synthetichealth/synthea/wiki/Keep-Patients-Module>

A.2.3. SYNTHETIC DATA PROCESSING

Synthea’s CSV output consists of the following major components: conditions.csv, patients.csv, and procedures.csv. Normally these csv files consist of generated patients’ life-long records. We first locate the unique patient ID, then use conditions.csv to locate a particular disease. We record the time, then use the patient ID, disease name, and time to locate the procedure code in procedures.csv. We run a batch of 50 patients per disease for our analysis.

Table 6: Medical Services Table

Medical Service Category	Synthea			LLMSYN			MIMIC		
	SNOMED CT Code	English Description	Count	CPT Code	English Description	Count	CPT Code	English Description	Count
Diagnostic Imaging Services	399208008	Plan X-Ray of Chest	45	71020 71250	Chest X-ray, two views CT scan of the chest without contrast	30 18			
Respiratory Support and Treatment	371908008 431182000	Oxygen administration Placing Patient in prone position	45 45	94640	Pressurized or non-pressurized inhalation treatment	23	94002 94003	Ventilatory management (initiation) Ventilatory management (subsequent days)	7 15
Emergency Care and Monitoring	315639002 385763009	Initial patient assessment Hospice care (regime/therapy)	26 26	99291 99213	Critical care, first hour Office or other outpatient visit, moderate complexity	19 25	99291 99292	Critical care, first hour Critical care, additional 30 minutes	23 10
Ward Management and Daily Care				99232 99233	Subsequent hospital care, daily management Subsequent hospital care, high complexity	24 15	99232 99233	Subsequent hospital care, daily management Subsequent hospital care, high complexity	19 19
Intervention and Surgical Procedures	310417005	Certification procedure	26	36556 31500	Insertion of non-tunneled centrally inserted central venous catheter Intubation, endotracheal, emergency procedure	23 18	36556 31624	Insertion of central venous catheter Bronchoscopy with biopsy	7 10
Planning and Coordinating Care	370789001	Development of individualized plan of care	26	99214	Office or other outpatient visit, complex	16			
Notification and Assessment	185087000	Notifications	26						
Consultation and Detailed Assessment							99253 99254 99255	Inpatient consultation, detailed Inpatient consultation, comprehensive Inpatient consultation, highly complex	5 7 10

A.3. Extensible Study: Prescription Generation

We performed a sample study for Prescription Generation. It is an extra step right after Step 4: CPT Code Generation. Since prescriptions are hard to evaluate, we left them here in the Appendix.

The following is the prompt for Prescription Generation:

You are a professional medical doctor with extensive experience in handling various complex medical situations and interacting with a large amount of medical data over more than twenty years. Now, I would like you to utilize your experience to manually provide me with a synthetic medical dataset based on reasoning. You are prohibited from using any coding tools.

Instructions

DISEASE NAME

[INSURANCE]: Fill in based on the patient's type of insurance.

[AGE]: Indicate the patient's age.

[LANGUAGE]: Indicate the patient's preferred [LANGUAGE].

[RELIGION]: Fill in according to the patient's religious beliefs.

[MARITAL_STATUS]: Fill in based on the patient's marital status.

[ETHNICITY]: Fill in based on the patient's ethnic background.

[ICD9_CODE]: ICD9 Code of the disease.

[HOSPITAL_EXPIRE_FLAG]: Flag 1 if patient died, 0 as living.

[CPT_CD]: Describing the caring CPT code that is performed to the patient, in order. Can have

multiple CPT_CD.

Now, you are given with the Patient's DISEASE NAME, [LANGUAGE], [RELIGION], [INSURANCE],

[MARITAL_STATUS], [ETHNICITY], [HOSPITAL_EXPIRE_FLAG] and [CPT_CD]. Please try to inference the

possible PRESCRIPTION that the patient may receive during a 14-day stay session at hospital. The

PRESCRIPTION should be formed by following categories:

A inference paragraph describing the details on how you inferred the usage of following drugs, why

in this order and why are the usage. Any corner cases? What should we keep in mind considering

patient's information, for instance, age, insurance, marital_status, etc?

LLMSYN

DRUG_TYPE: This indicates the category or type of drug prescribed, such as a main medication (MAIN)
↪ or a base solution (BASE) like D5W (5% Dextrose in Water).

DRUG: The name of the medication prescribed (e.g., Tacrolimus, Warfarin).

FORMULARY_DRUG_CD: DRUG's coding(abbreviation).

DOSE_VAL_RX: The dose value prescribed, for example, '2' mg or '25000' units.

DOSE_UNIT_RX: The unit of the dose prescribed, such as mg (milligrams), ml (milliliters), units, etc.

FORM_VAL_DISP: The form value dispensed, which might indicate the quantity of the medication
↪ dispensed, like '2' capsules or '1' bag.

FORM_UNIT_DISP: The unit of the form dispensed, for example, CAP (capsule), BAG, TAB (tablet), etc.

ROUTE: This indicates the route of administration for the drug, such as PO (orally), IV
↪ (intravenously), SC (subcutaneously), etc.

All PRESCRIPTION's DRUGS serves for helping to relieve the disease. Make sure the DRUG are generated
↪ in order, you will have to fill out their perform series as well. MAKE SURE YOUR OUTPUT IS IN THE
↪ FORMAT OF TABLE.

Make sure your process is created by inference. Make sure you generate a final output part, like the
↪ following example.

GIVEN PATIENT INFO:

Here is a sample generation with GPT-4:

To create a synthetic medical dataset based on the provided parameters and infer a prescription for a
↪ hypothetical patient during a 14-day hospital stay, let's proceed through the following steps:

Patient Information

Given the parameters provided:

ADMISSION_TYPE: EMERGENCY

INSURANCE: Medicare

LANGUAGE: English

RELIGION: Catholic

MARITAL_STATUS: Married

ETHNICITY: White

HOSPITAL_EXPIRE_FLAG: 0 (the patient did not expire)

ICD9_CODE: 250.11 (Type 1 Diabetes Mellitus, uncontrolled)

Inference on Prescriptions

Type 1 Diabetes Mellitus, particularly when uncontrolled, requires careful management of blood

↪ glucose levels primarily through insulin therapy. Given the patient's context and the emergency

↪ nature of the admission, the following prescription regimen can be inferred:

Insulin Administration: Primary treatment for Type 1 Diabetes, to manage blood glucose levels.

Electrolytes: Necessary if the patient experiences electrolyte imbalances due to hyperglycemia.

Antihypertensives: Often required if the patient has diabetes-associated hypertension.

D5W: As a base solution to aid in the delivery of intravenous medications and manage hydration.

Inference Explanation

Insulin Glargine and Lispro: Used in a basal-bolus regimen for tight glucose control. Glargine as a
↪ long-acting basal insulin, and Lispro for mealtime blood sugar control.

D5W: A dextrose solution, which can help modulate blood glucose levels and provide a medium for
↪ delivering soluble drugs.

Potassium Chloride: High blood glucose can cause diuresis and potassium loss; thus, supplementation
↪ is necessary.

Lisinopril: Often used in diabetic patients to manage hypertension and protect kidney function.

LLMSYN

DRUG_TYPE	DRUG	FORMULARY_DRUG_CD	DOSE_VAL_RX	DOSE_UNIT_RX	FORM_VAL_DISP	FORM_UNIT_DISP	ROUTE
MAIN	Insulin Glargine	INGLA	30	Units	1	Vial	SC
MAIN	Insulin Lispro	INLIS	10	Units	As needed	Vial	SC
BASE	D5W	D5W500	500	ml	1	Bag	IV
MAIN	Potassium Chloride	KCL20	20	meq	1	Bag	IV
MAIN	Lisinopril	LIS5	5	mg	1	Tab	PO

Figure 5: Sample Prescription Table.

A.4. Additional Results

To further assess the quality of synthetic electronic health records (EHRs) generated by LLMSYN, we conducted an in-depth evaluation of their distribution characteristics. Our analysis focused on two key aspects: the mortality rate and the joint distribution of ICD9 diagnostic codes.

Notice that INJECT refers to $\text{LLMSYN}_{\text{full}}$, CLEAN refers to $\text{LLMSYN}_{\text{base}}$, MIMIC refers to MIMIC-III’s distribution.

For the evaluation of ICD-9 code distributions, we employed the following methodology:

1. We identified the 15 most frequently occurring ICD9_Codes in the MIMIC dataset.
2. Using this set of codes as a reference, we analyzed the distribution of these top 15 codes in our synthetic data.
3. We then compared this distribution to the distribution of all other ICD9_Codes present in the synthetic dataset.

This approach allowed us to assess how well LLMSYN-pipeline captures both the most common diagnostic patterns and the overall diversity of medical conditions represented in the synthetic EHRs.

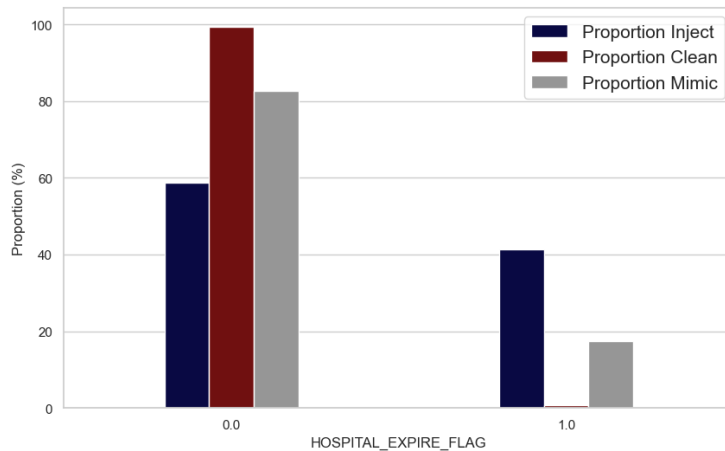


Figure 6: $\text{LLMSYN}_{\text{base}}$, model=LLaMA, Synthetic Death Rate.

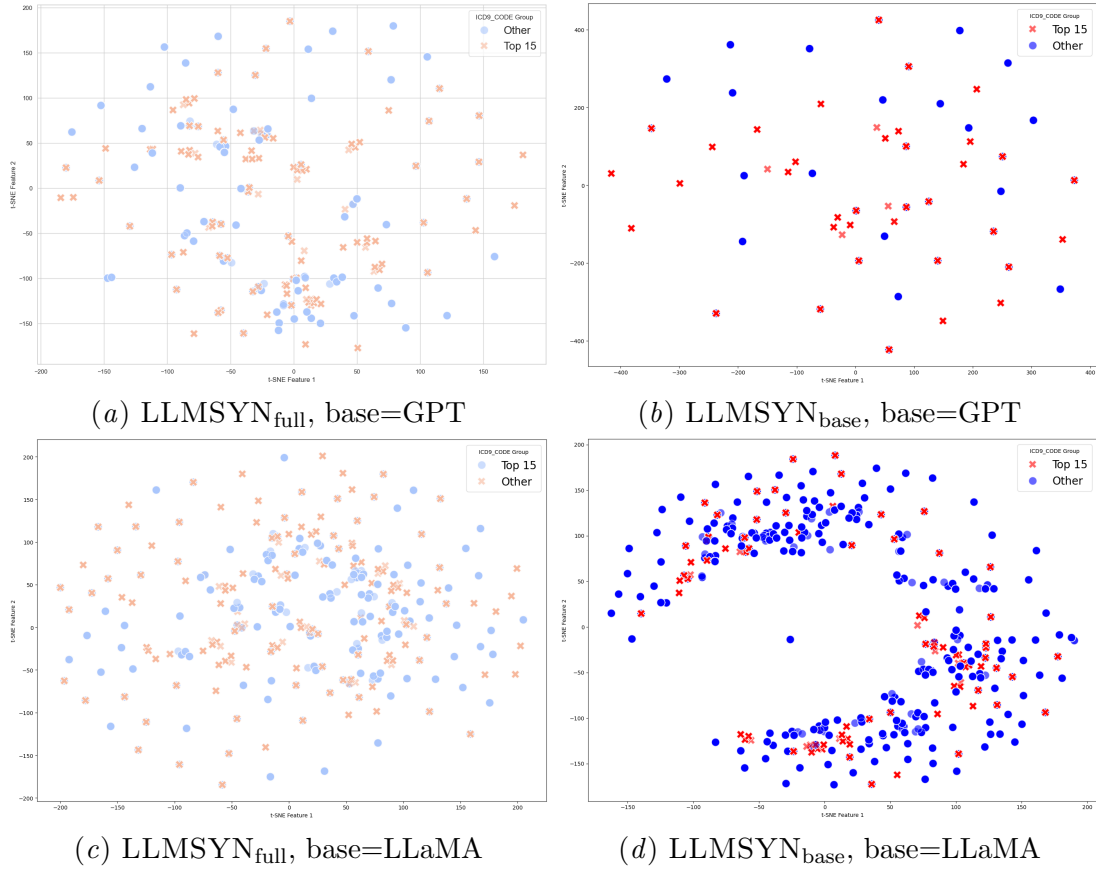


Figure 7: Synthetic ICD9_CODE Distribution (Top-15 vs Others) for different LLM-SYN variants.

A.5. Synthetic ICD9_CODE Proportion

Here, we first filtered MIMIC-III’s Top-15 ICD9_CODE, then compared MIMIC-III Top-15 ICD9_CODE distribution with LLMSYN’s synthetic ICD9_CODE distribution.

Notice that INJECT refers to LLMSYN_{full}, CLEAN refers to LLMSYN_{base}, MIMIC refers to MIMIC-III’s distribution. It can be observed that LLMSYN_{full} generates a more diverse range of ICD9_CODE, closely mirroring the real data. This demonstrates the effectiveness of injecting additional knowledge into the generation process.

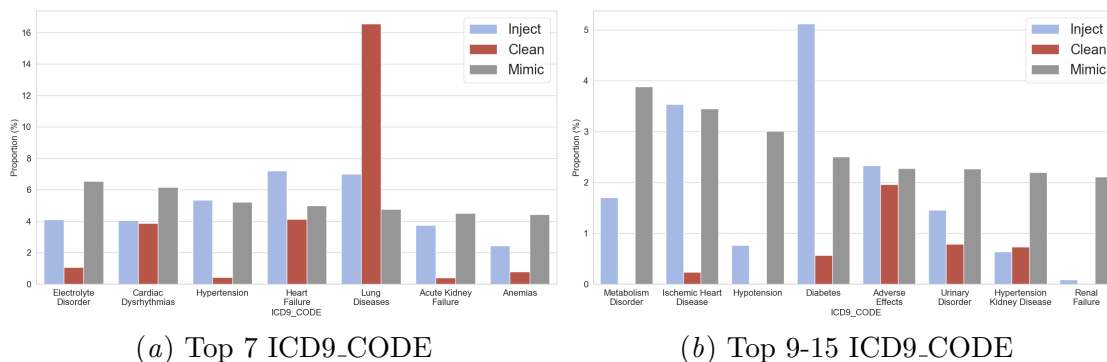


Figure 8: model=GPT, Synthetic ICD9_CODE Proportion.

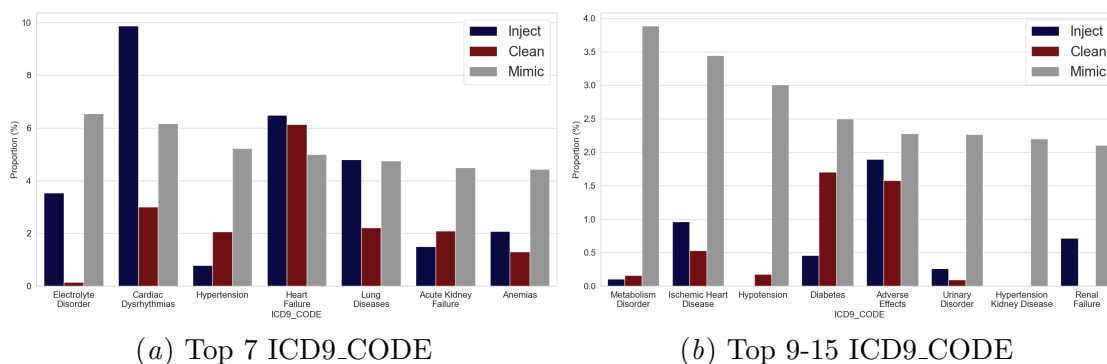


Figure 9: model=LLaMA, Synthetic ICD9_Code Proportion.