PromptEHR: Conditional Electronic Healthcare Records Generation with Prompt Learning

Zifeng Wang and Jimeng Sun
University of Illinois Urbana-Champaign
{zifengw2, jimeng}@illinois.edu

Abstract

Accessing longitudinal multimodal Electronic Healthcare Records (EHRs) is challenging due to privacy concerns, which hinders the use of ML for healthcare applications. Synthetic EHRs generation bypasses the need to share sensitive real patient records. However, existing methods generate single-modal EHRs by unconditional generation or by longitudinal inference, which falls short of low flexibility and makes unrealistic EHRs. In this work, we propose to formulate EHRs generation as a text-to-text translation task by language models (LMs), which suffices to highly flexible event imputation during generation. We also design prompt learning to control the generation conditioned by numerical and categorical demographic features. We evaluate synthetic EHRs quality by two perplexity measures accounting for their longitudinal pattern (longitudinal imputation perplexity, `lpl`) and the connections cross modalities (cross-modality imputation perplexity, `mpl`). Moreover, we utilize two adversaries: membership and attribute inference attacks for privacy-preserving evaluation. Experiments on MIMIC-III data demonstrate the superiority of our methods on realistic EHRs generation (53.1% decrease of `lpl` and 45.3% decrease of `mpl` on average compared to the best baselines) with low privacy risks.

1 Introduction

Electronic healthcare records (EHRs) fuel the development of machine learning models for healthcare applications (Wang et al., 2021a,b; Choi et al., 2016b,a). However, sharing EHR data usually undergoes strict and expensive de-identification and administration processes thus being difficult. Although there have been attempts on perturbing potentially identifiable attributes as the de-identification step (Emam et al., 2015), they were argued not immune to the hack for re-identification (El Emam et al., 2011; Choi et al., 2017). Alternatively, generating synthetic but realistic EHRs can circumvent data leakage while preserving the patterns of real EHRs for further research and development (Biswal et al., 2020).

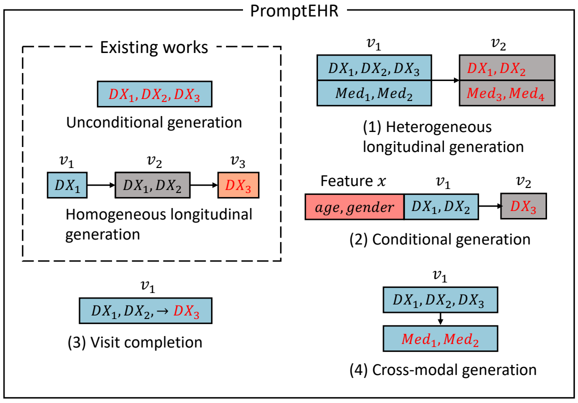


Figure 1: A conceptual demonstration of how `PromptEHR` works more flexible than all existing works. v_t indicates the t-th visit; *DX*, *Med* are short for diagnosis and medication events; Red are the targets to generate. Our method (`PromptEHR`) is amenable to four new conditional generation ways thus more controllable and flexible.

Deep generative models like GANs (Goodfellow et al., 2014) and VAEs (Kingma and Welling, 2013) have become popular for unconditional EHRs generation (Choi et al., 2017) and longitudinal EHRs generation (Biswal et al., 2020; Zhang et al., 2020) for diagnosis codes. However, EHRs are often multimodal with different types of events, including diagnoses, procedures, medications, and also patient baseline demographic features like age and gender (Johnson et al., 2016). GANs & VAEs usually struggle to model complex multimodal and non-Gaussian distributions as well as sparse one-hot-encoded vectors (Xu et al., 2019). By contrast, generative language models (LMs) are proved highly powerful to represent large and complex distributions on discrete data (e.g., texts) (Liu et al., 2021b; Radford et al., 2021), which makes them promising for EHRs generation.

In this work, we propose to leverage generative language models (LMs) for EHRs generation. We try to generate a sequence of visits with mixed types of events, e.g., diagnosis and medications. As Fig. 1 shows, previous works make unconditional generation for single-modal static EHRs (Choi et al., 2017) or for single-modal longitudinal EHRs (Zhang et al., 2021). However, real EHRs are *heterogeneous* with multiple types of temporal events and have baseline patient features, e.g., demographic information. We seek to (1) generate realistic mixed-type longitudinal EHRs with scale and (2) support flexible conditional generation to fit the need for personalized EHRs. Specifically, our contributions are

- We propose a new EHRs generation method making the best of LMs, which enables generating multimodal EHRs.
- We design prompt learning for controllable and flexible EHRs generation with LMs.
- We design comprehensive evaluation for both quality and privacy of the generated EHRs.

2 Related Works

2.1 EHRs Generation

Early works on generating EHRs (Lombardo and Moniz, 2008; Buczak et al., 2010; McLachlan et al., 2016) are rule-based methods. However, they were argued not capable of providing realistic data for machine learning tasks and were still vulnerable to re-identification (Choi et al., 2017). Deep generative models advanced by the power of deep learning, e.g., variational auto-encoders (VAE) (Kingma and Welling, 2013) and generative adversarial network (GAN) (Goodfellow et al., 2014), gained most attention recently. Choi et al. (2017) pioneered in adapting GAN for discrete patient records generation, namely MedGAN, which was followed by improving GANs for EHRs generation (Guan et al., 2018; Baowaly et al., 2019; Zhang et al., 2020); using VAE (Biswal et al., 2020), hybrid GANs (Lee et al., 2020; Cui et al., 2020), or conditional GANs (Xu et al., 2019). However, most methods only generate static tabular EHRs or longitudinal single-modal EHRs. GANs are often riddled with mode collapse, non-convergence, and instability, which cause their training tricky in practice (Saxena and Cao, 2021). Moreover, due to the representation limit, GANs struggle in modeling multimodal distributions and sparse one-hot-encoded vectors (Xu et al., 2019) while EHRs are with these properties. By contrast, we bypass these challenges by LMs.

2.2 Language Models & Prompt Learning

LMs are often used for text generation tasks attributed to their *auto-regressive* nature, e.g., T5 (Raffel et al., 2020) and BART (Lewis et al., 2020). Nonetheless, they cannot be directly applied to EHRs generation since EHRs consist of not only plain clinical notes but also longitudinal sequences of events. Although there were works on generating medical texts by LMs (Amin-Nejad et al., 2020; Libbi et al., 2021; Kagawa et al., 2021), none has been done for synthetic EHRs generation. Prompt learning was used to control the topic of text generation (Li and Liang, 2021; Yu et al., 2021; Qian et al., 2022). However, they only consider one-hot encoded topics as prefix. In this work, we leverage prompt learning for EHRs generation conditioned on patient baseline features, which include both categorical and numerical values.

3 Methods

In this section, we elaborate on the main framework of `PromptEHR`, including the problem setting, workflow, and training tasks formulation. Next, we discuss the strategies for generating diverse synthetic EHRs with minor loss of quality. Then, we present the recipe proposed for the evaluation for both quality and privacy-preserving ability of the EHRs generation models.

3.1 Problem Formulation

Consider there are N patients where the n-th patient is represented by $X_{n,1:T_n} = \{\mathbf{x}_n; \mathbf{x}_{n,1}, \mathbf{x}_{n,2}, \ldots, \mathbf{x}_{n,T_n}\}$ where $\mathbf{x}_n$ are the baseline features, e.g., age and gender; $\mathbf{x}_{n,t}$ signifies events happened at the t-th visit; T_n is the total number of visits. For each visit $\mathbf{x}_{n,t}$, we have K types of events as $\mathbf{x}_{n,t} = \{\mathbf{x}^1_{n,t}, \mathbf{x}^2_{n,t}, \ldots, \mathbf{x}^K_{n,t}\}$. $\mathbf{x}^k_{n,t} = \{c_1, c_2, \ldots, c_l\}$ are all events of type k, l is the number of events.

We formulate three basic functions to support EHRs generation:

- **Longitudinal imputation**: given historical visits $X_{n,1:t} = \{\mathbf{x}_{n,1}, \ldots, \mathbf{x}_{n,t}\}$, the model predicts the events in next visit as $\mathbf{x}_{n,t+1}$;
- **Cross-modality imputation**: given visits with $K-1$ types of events $\mathbf{x}_{n,t} \setminus \{\mathbf{x}^k_{n,t}\}$, the model predicts the events belonging to modality k;

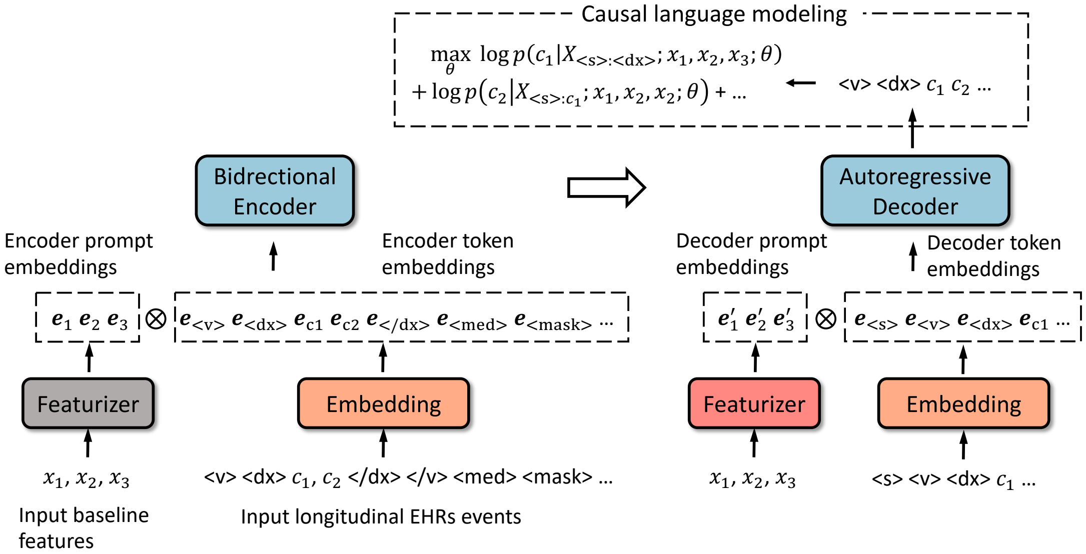

Causal language modeling
$\max_{\theta} \log p(c_1|X_{<s>:<dx>}; x_1, x_2, x_3; \theta)$
$+ \log p(c_2|X_{<s>:c_1}; x_1, x_2, x_2; \theta) + \ldots$
<v> <dx> c_1 c_2 ...
Bidirectional Encoder
Autoregressive Decoder
Encoder prompt embeddings
Encoder token embeddings
Decoder prompt embeddings
Decoder token embeddings
e_1 e_2 e_3 ⊗ $e_{<v>}$ $e_{<dx>}$ e_{c1} e_{c2} $e_{</dx>}$ $e_{<med>}$ $e_{<mask>}$ ...
e'_1 e'_2 e'_3 ⊗ $e_{<s>}$ $e_{<v>}$ $e_{<dx>}$ e_{c1} ...
Featurizer
Embedding
Featurizer
Embedding
x_1, x_2, x_3
<v> <dx> c_1, c_2 </dx> </v> <med> <mask> ...
x_1, x_2, x_3
<s> <v> <dx> c_1 ...
Input baseline features
Input longitudinal EHRs events

Figure 2: The workflow of PromptEHR. The input longitudinal events are transformed to the code sequence by special tokens, e.g., <v> and </v> cover events in the same visit; <dx> and </dx> cover contemporary diagnosis events. Baseline features are encoded to prompt embeddings by two *featurizers* then add to the token embeddings. The model decodes autoregressively and is trained with causal language modeling loss.

- **Conditional generation**: given historical visits $X_{n,1:t}$ and the baseline features $\mathbf{x}_n$, the model makes further predictions.

These functions can be combined to synthesize EHRs from the existing partial EHRs with baseline features or from scratch.

3.2 Encoding

The overview is shown by Fig. 2. The first step is to transform the raw inputs $X_{n,1:T_n}$ to token sequences hence acceptable to the encoder.

Inputs tokenization. PromptEHR is compatible with all sequence-to-sequence models (Cho et al., 2014). We choose to utilize BART (Lewis et al., 2020) as the base model. BART uses a bidirectional encoder thus allowing arbitrary corruption for the input sequences and a left-to-right decoder to reconstruct the inputs. Motivated by the application of prompts in language (Liu et al., 2021a), we leverage prompts to specify the inputs. Without loss of generality, we assume two modalities: diagnosis (DX) and medication (Med). Denote [X] and [Z] as the input and answer slots, we can formulate the longitudinal imputation task by a *prefix prompt* problem: <v>[X]</v>[Z]. The model tries to fill the answer slot [Z] which are the events in the next visit; the cross-modal imputation task is built by a *cloze prompt* problem: [X]<dx>[Z] where <dx> signifies the start of diagnosis events and [X] represents the multimodal context events.

Conditional prompt featurizer. We introduce *conditional prompt embeddings* to enable conditional generation based on patient features. We consider both *categorical* $\mathbf{x}_{cat}$ and *numerical* features $\mathbf{x}_{num}$. The categorical prompt embeddings $\mathbf{E}_{cat}$ is obtained by

$$\mathbf{E}_{cat} = (\mathbf{x}_{cat}\mathbf{W}_0 + \mathbf{b})\mathbf{W}_1. \qquad (1)$$

$\mathbf{x}_{cat}$ has m_c mulit-hot encoded indices indicating the classes of each feature; $\mathbf{W}_0 \in \mathbb{R}^{m_c \times d_0}$; $\mathbf{W}_1 \in \mathbb{R}^{d_0 \times d_1}$. Therefore, $\mathbf{e}_{cat}$ encodes the instruction of $\mathbf{x}_{cat}$ and steers the LM to generate specific populations. We transform $\mathbf{x}_{num} \in \mathbb{R}^{m_u}$ to $\mathbf{e}_{num}$ with another set of $\mathbf{W}_0$, $\mathbf{W}_1$, and $\mathbf{b}$. $\mathbf{E}_{cat}$ and $\mathbf{E}_{num}$ then prepend to token embeddings by

$$\mathbf{E} = [\underbrace{\mathbf{E}_{cat}; \mathbf{E}_{num};}_{\text{Prompt Embeddings}} \mathbf{E}_{tok}] \qquad (2)$$

to serve as the inputs to the encoder. We build the inputs for the decoder with the other featurizer to get $\mathbf{E}'_{cat}$ and $\mathbf{E}'_{num}$ and the shared token embeddings $\mathbf{E}_{tok}$.

3.3 Decoding & Training

The inputs tokens for the decoder are shifted encoder inputs such that the decoder predicts the next token based on the prior tokens. Denote the context by $\mathbf{X}$ and the target event by $\mathbf{x}$, the true conditional distribution is $p(\mathbf{x}|\mathbf{X})$. For instance, in the longitudinal imputation task, the context is the historical record of the patient $\mathbf{X}_{1:t}$ and the target is the events in the next visit $\mathbf{x}_{t+1}$. Correspondingly, $p(\mathbf{x}|\mathbf{X};\theta)$ is the prediction made by the model. We use $\tilde{\mathbf{X}} \sim q(\mathbf{X})$ to represent the perturbations added to the context inputs. The training objective is to minimize the negative log-likelihood as

$$\mathcal{L} = \mathbb{E}_{\mathbf{X}\sim p(\mathbf{X})}\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}|\mathbf{X})}\mathbb{E}_{\tilde{\mathbf{X}}\sim q(\mathbf{X})}[-\log p(\mathbf{x}|\tilde{\mathbf{X}};\theta)]. \tag{3}$$

The model is hence pushed to maximize the predicted probability to the true next tokens $\mathbf{x}$ conditioned by the corrupted inputs $\tilde{\mathbf{X}}$.

We apply the following corruptions during training: (1) Token mask, infill, and deletion; (2) Span shuffle and permutation. For (1), we randomly replace multiple tokens with <mask> or delete as length $\sim$ Poisson(3). For (2), we randomly shuffle the tokens within the same visits and shuffle the modality orders in the same visits.

3.4 Harmless Randomness in Generation

Apart from preciseness, the *diversity* of the generated data is also of great importance. PromptEHR samples from the conditional distribution by

$$\mathbf{x} \sim p(\mathbf{x}_t|X_{1:t-1};\theta), \tag{4}$$

which allows to adjust diversity by many techniques existing in natural language generation literature. For instance, to prevent low probability events, we can apply *top-k* sampling (Fan et al., 2018). Temperature is also useful to flatten or sharpen the conditional distribution. More advanced methods, e.g., beam search (Welleck et al., 2019) and nucleus sampling (Holtzman et al., 2019) are all available for exploitation by PromptEHR, which brings a great potential to achieve higher quality EHRs with diversity. By contrast, GANs & VAEs depend on sampling random noise vectors to introduce diversity, which is not controllable and usually undermines generation quality.

3.5 Quality Evaluation

We provide a recipe to evaluate EHRs generation on two dimensions: **accuracy** and **privacy**. For accuracy, we propose to adopt perplexity which is usually used in the text generation task, defined by the exponent of the average negative log-likelihood (NLL) per word (Neubig, 2017):

$$\texttt{ppl} = e^{-(\log \prod_{l=1}^{L} p(c_l|c_{1:l-1};\theta))/L}, \tag{5}$$

where $p(v_l|v_{1:l-1})$ indicates how the model predicts the next word using all previous words as the context; L is the length of the document; θ is the model parameter. Intuitively, a random predictor will produce ppl that is equal to the cardinality of vocabulary $|\mathcal{C}|$. We hereby adapt it to the longitudinal imputation perplexity (lpl) and cross-modality imputation perplexity (mpl) taking the structure of EHR into account.

lpl takes the *temporal coherence* of the patient visits into account. For instance, chronic diseases like diabetes can cause complications (e.g., heart disease and kidney failure) in the future. Following Eq. (5), we can write the lpl of a patient's records $X = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$ as

$$\begin{aligned}\texttt{lpl} &= e^{-\sum_{t=1}^{T} \log P(\mathbf{x}_t|\mathbf{x}_{1:t-1};\theta)/(l_t * T)} \\ &= e^{-\sum_{t=1}^{T}\sum_{l=1}^{l_t} \log P(v_l|\mathbf{x}_{1:t-1};\theta)/(l_t * T)}.\end{aligned} \tag{6}$$

Here, $\mathbf{x}_t = \{c_1, \dots, c_{l_t}\}$ are all events during the t-th admission. Inside this admission, concurrent events are independently generated conditioned on previous visits, therefore we can decompose $p(\mathbf{x}_t|\mathbf{x}_{1:t-1};\theta) = \prod_{l=1}^{l_t} p(c_l|\mathbf{x}_{1:t-1};\theta)$ then come to the results.

mpl accounts for the correlations between modalities. For example, high body temperature in lab test may correspond to fever in diagnosis. We focus on the t-th admission where the joint distribution of all K modalities $p(\mathbf{x}_t^1, \dots, \mathbf{x}_t^K|\mathbf{x}_{1:t-1};\theta)$. We can write the NLL here by

$$\begin{aligned}\text{NLL}_t &= -\frac{1}{K}\sum_{k=1}^{K} \log p(\mathbf{x}_t^k|\mathbf{x}_t^{1:K\setminus k}, \mathbf{x}_{1:t-1};\theta) \\ &= -\frac{1}{K}\sum_{k=1}^{K}\frac{1}{l_t^k}\sum_{l=1}^{l_t^k} \log p(v_l|\mathbf{x}_t^{1:K\setminus k}, \mathbf{x}_{1:t-1};\theta),\end{aligned} \tag{7}$$

where l_t^k indicates the number codes belonging the k-th modality. Next, we can track all admissions to obtain the final definition of mpl by

$$\texttt{mpl} = e^{\sum_{t=1}^{T} \text{NLL}_t/T}. \tag{8}$$

3.6 Privacy Evaluation

It is crucial to measure the privacy preserving when sharing the synthetic data. We try to evaluate two privacy risks: **membership inference** and **attribute inference**. We split the data into the training data $\mathcal{D}_1 = \{X_{n,1:T_n}\}_{n=1}^{N}$ and testing data $\mathcal{D}_2$, and generate synthetic data $\mathcal{D}_S$ with the same length as $\mathcal{D}_1$.

Membership Inference. Attackers would try to infer the membership of the patient records based on the real records they own. We design this adversary based on shadow training (Shokri et al., 2017). In the first stage, a shadow model M_{sd} is trained on $\mathcal{D}_S$. It tries to mimic the performance of the generation model in longitudinal inference.

In the second stage, a membership inference dataset is built based on $M_{\text{sd}}(X)$ where $X \in \widetilde{\mathcal{D}}_S \bigcup \mathcal{D}_2$. $\widetilde{\mathcal{D}}_S$ is a subset of $\mathcal{D}_S$ with the same number as $\mathcal{D}_2$. A model $M_{\text{mi}} : \mathbb{Y}_{\texttt{ppl}} \mapsto \{0, 1\}$ is trained to differentiate if X comes from $\mathcal{D}_S$ or $\mathcal{D}_2$. We will then evaluate the success rate of M_{mi} on identifying $X \in \mathcal{D}_1 \bigcup \mathcal{D}_2$. The better the adversary $M_{\text{sd}}(X)$ and M_{mi} perform on this evaluation, the higher the privacy risk caused by releasing the synthetic EHRs.

Attribute Inference. We build this adversary following (Zhang et al., 2021). In this case, attackers hold some incomplete real records where several sensitive attributes are missing. They would take advantage of the synthetic data to infer these attributes. Besides, attackers also hold the prior knowledge of association between the attributes, i.e., given the incomplete individual records, how probable another code appears in expectation or $P_0 = p(v_l | \{v_1, \ldots, v_{l_t}\}_{t=1}^T \setminus v_l)$. With the prior, the attacker will train an attribute imputation model on the synthetic data $\mathcal{D}_S$, i.e., $\hat{P} = p(v_l | \{v_1, \ldots, v_{l_t}\}_{t=1}^T \setminus v_l; \theta_I)$. The attacker then believe the code v_l exists when $\log \hat{P} - \log P_0 \geq \delta$. δ is a pre-defined threshold. In experiments, we train another attribute imputation model on $\mathcal{D}_1$ to approximate the prior knowledge. We evaluate the success rate of this attack. Besides, we create a control arm where another imputation model is trained on the test set. Comparison between the control and the treatment (imputation model trained on $\mathcal{D}_S$) suffices for an immediate evaluation of the synthetic data's risk level.

4 Experiments

In this section, we designed experiments to answer the following questions.

- **Q1.** How well does `PromptEHR` perform for EHRs generation compared with the state-of-the-art methods on generation quality?
- **Q2.** What is the level of privacy risk on membership inference and attribute inference of the generated EHRs by `PromptEHR`?
- **Q3.** Are the synthetic data useful for the secondary use by predictive modeling in practice?

Table 1: Statistics of the used MIMIC-III data.

Item	**Number**	**Event Type**	**Number**
Patients	46,520	Diagnosis	1,071
Total Visits	58,976	Drug	500
Total Events	5,401,961	Procedure	668
Events per Patient	116	Lab Test	185

- **Q4.** How is the generation quality of `PromptEHR` influenced by the size of training records?

4.1 Experimental Setup

Dataset. (Johnson et al., 2016) We use MIMIC-III data which has 46k patients' records collected from the intensive care unit. We pick the diagnosis, procedure, drug, and lab test as the target events for generation. All events in the same admission are seen as contemporary. We randomly split the 46,520 patients records into 39,581, 2,301, 4,633 for the train/validation/test set. The data statistics are available in Table 1.

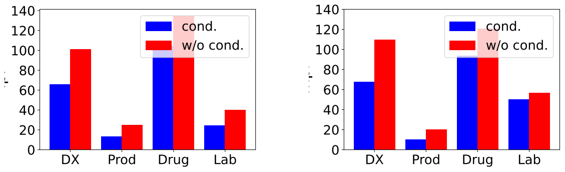
Baselines. We compare the following baselines:

- LSTM+MLP. This is the baseline that leverages LSTM (Hochreiter and Schmidhuber, 1997) to learn the patient state thus extracting the temporal visit patterns. Based on the state embeddings, MLP layers are able to impute the probability of events within the visit or for the next visit.

- LSTM+MedGAN (Choi et al., 2017). The original MedGAN is not able to do conditional generation and temporal inference. Similar to the first baseline, LSTM is used for capturing temporal patterns as the inputs for MedGAN. Then, the generator of MedGAN will try to make conditional generation for records as realistic as possible to fool its discriminator.

- SynTEG (Zhang et al., 2021). This is one of the most recent EHRs generation methods. It also consists of a state embedding module and a imputation module. It utilizes transformers (Vaswani et al., 2017) for temporal dependency learning and conditional Wasserstein GAN with gradient penalty (WGAN-GP) (Arjovsky et al., 2017; Gulrajani et al., 2017) for event inference.

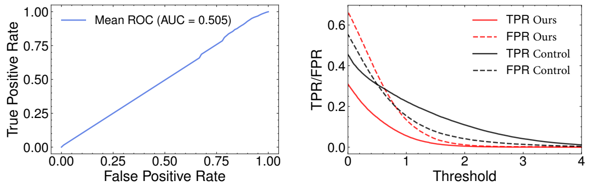
- GPT-2 (Radford et al., 2019). We pick GPT-2 as the LM baseline that only does causal



(a) Perplexity (`lpl`).

(b) Perplexity (`mpl`).

Figure 3: Perplexity compared between generation w/ (cond.) and w/o conditional prompts (w/o cond.) for four types of events. Note that both `lpl` and `mpl` are the less the better.



(a) The ROC curve of the membership inference attack by shadow training.

(b) The true positive rate (TPR) and false positive rate (FPR) of the attribute inference attack w.r.t. different thresholds δ.

Figure 4: Privacy-preserving evaluation on membership inference (left) and attribute inference (right) adversaries. On the right, the `PromptEHR` curves indicate the results of attribute inference model trained on the synthetic data $\mathcal{D}_S$ by `PromptEHR`; the Control curves indicate the one trained on test set $\mathcal{D}_2$.

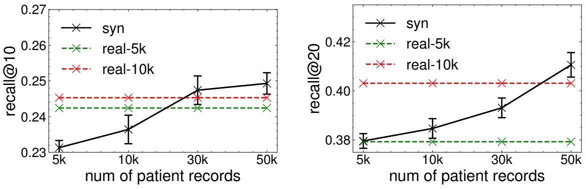
language modeling on EHRs. Then, it is able to do event generation like texts generation.

4.1.1 Evaluation metrics

We use the proposed `lpl` and `mpl` to evaluate generation quality. Since perplexity of different patient records vary significantly, we take the median of perplexity across patients for the sake of stability of the performance estimate.

We use two adversaries: membership inference (MI) and attribute inference (AI), to test the privacy risk. In MI, we use LSTM+MLP as the shadow model to mimic the outputs of `PromptEHR`. A three-layer MLP predicts the membership. ROC curve is plotted to evaluate the attack success rate; In AI, we train an LSTM+MLP on $\mathcal{D}_1$ to approximate the prior and another LSTM+MLP on $\mathcal{D}_S$ as the attribute imputation model.

To test the utility of the synthetic data for downstream predictive tasks, we train LSTM+MLP on $\mathcal{D}_S$ or $\mathcal{D}_2$ and test it on $\mathcal{D}_2$ to compute the recall@20/30.



(a) Recall@10.

(b) Recall@20.

Figure 5: Recall@10/20 of the predictive model on the test set with varying input data size: *syn* indicates the model trained on *fully synthetic data*; *real-5k/10k* indicate trained on 5k/10k real data. Error bars show the 95% confidence interval which also appear in the following figures.

4.2 Implementation Details

All the used LSTM+MLP model consists of a three-layer bi-directional LSTM with 128 hidden dimensions with one 256-dim MLP layer. It is trained with 1e-4 learning rate by Adam optimizer (Kingma and Ba, 2014). The 12-layer transformer based pre-trained GPT-2 is trained with 1e-5 learning rate and 1e-4 weight decay by Adam. We follow the architecture and training protocol from the original papers of MedGAN and SynTEG.

For `PromptEHR`, we use BART model as the backbone (Lewis et al., 2020). We use Adam by setting learning rate as 1e-5, weight decay as 1e-4, batch size as 16. The total training epoch is 50 where the first 3 epochs are warm-up steps. During the training stage, the perplexity computed on the validation set is used to pick the best checkpoint. All experiments are conducted with an RTX-3090 GPU, 251 GB RAM, and AMD Ryzen Threadripper 3970X 32-core CPU.

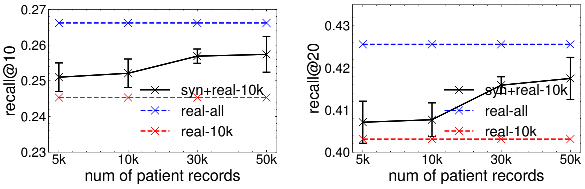
4.3 Q1. Generation Quality

The calculated `mpl` and `lpl` of all show in Table 2. It is witnessed that `PromptEHR` obtains the best result among all methods. On the contrary, LSTM+MedGAN and SynTEG do not gain better test perplexity than the basic LSTM+MLP. The main reason is that their GAN part takes a noise input except for the learned temporal state embeddings to make conditional generation. GPT-2 works better than LSTM+MLP on temporal perplexity crediting to its power in capturing series pattern through transformers.

Most methods obtain better `mpl` than `lpl`. It is intuitive because models know the additional in-visit information from the other modalities for the target modality imputation, thus making better predictions. However, GPT-2 performs worse on

Table 2: Longitudinal imputation perplexity (`lpl`) & cross-modality imputation perplexity (`mpl`) of models on different kinds of events. Best values are in bold. ± value indicates the 95% confidence interval.

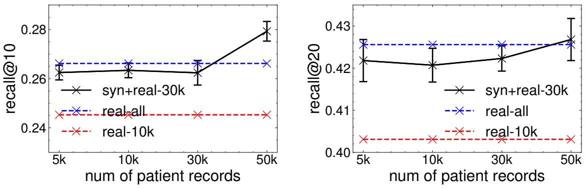
Method/Event	**Diagnosis**		**Procedure**		**Drug**		**Lab Test**	
perplexity	`lpl`	`mpl`	`lpl`	`mpl`	`lpl`	`mpl`	`lpl`	`mpl`
LSTM+MLP	125.1 ± 5.3	122.9 ± 2.0	40.3 ± 1.7	43.8 ± 0.9	173.3 ± 1.9	169.5 ± 0.5	68.9 ± 0.3	71.3 ± 0.5
LSTM+MedGAN	169.2 ± 6.0	109.8 ± 3.1	54.4 ± 2.5	40.1 ± 1.4	197.3 ± 2.5	166.7 ± 0.9	76.9 ± 0.3	66.2 ± 0.2
SynTEG	130.4 ± 4.6	130.0 ± 2.6	46.4 ± 1.8	46.2 ± 1.5	175.6 ± 2.0	175.4 ± 0.9	69.5 ± 0.2	69.6 ± 0.3
GPT-2	121.1 ± 1.8	134.2 ± 0.9	38.7 ± 0.9	48.2 ± 0.5	166.4 ± 1.8	169.6 ± 0.6	69.7 ± 0.1	69.6 ± 0.1
`PromptEHR`	**65.9 ± 2.0**	**67.7 ± 0.6**	**13.5 ± 0.8**	**10.1 ± 0.3**	**104.7 ± 1.8**	**93.7 ± 0.5**	**24.4 ± 0.1**	**50.1 ± 0.1**



(a) Recall@10.

(b) Recall@20.

Figure 6: Recall of the predictive model on the test set with varying input data size: *syn+real-10k* indicates the model trained on the *hybrid of synthetic & 10k real data*; *real-10k/all* indicate trained on 10k/all real data.



(a) Recall@10.

(b) Recall@20.

Figure 7: Recall of the predictive model on the test set with varying input data size: *syn+real-30k* indicates the model trained on the *hybrid of synthetic & 30k real data*; *real-30k/all* indicate trained on 30k/all real data.

`mpl` than on `lpl`. GPT-2 is trained by causal language modeling task where it models the sequence autoregressively. Without the prompt design, it is confused by the order of events within the same visit, which induces deteriorating performance.

Fig. 3 demonstrates the comparison made between generation w/ and w/o conditional prompts for `PromptEHR`. We identify that conditional prompts significantly improve the generation quality as they provide important characteristics of the patients. We are hence able to generate for specific populations with input prompts.

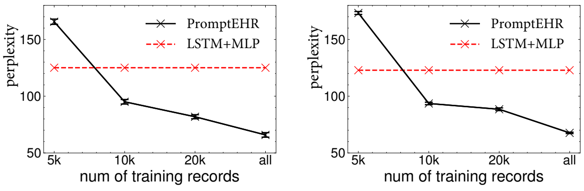
4.4 Q2. Privacy Evaluation

We test the privacy preserving ability of the generated synthetic EHRs by applying membership and attribute inference attacks. Results are illustrated by Fig. 4. Fig. 4a demonstrates the ROC curve consisting of true positive rate (TPR) and false positive rate (FPR) of the membership inference on $\mathcal{D}_1 \bigcup \mathcal{D}_2$. It clearly shows the MI model has bad performance that is near random guess (AUC $\simeq$ 0.5), which means the MI attack gains no sensitive membership information when trained on the synthetic data $\mathcal{D}_S$.

Fig. 4b shows the TPR/FPR of attribute inference attack based on shadow training with the varying threshold δ. Here, we cut the curve where $\delta = 4$ because all the remaining curves are approaching zero on its right. The threshold δ adjusts to the confidence level of the attacker, i.e., the smaller δ is set, the higher probability that the AI is correct we believe. When $\delta = 0$, so long as the AI inference probability $P(v_l)$ is larger than the prior $P_0(v_l)$, the AI model will believe the attribute v_l exists. In this scenario, both two models have a high FPR of around 0.6, but the TPR of `PromptEHR` is only near half of the control model. The TPR then keeps a much lower level when δ increases, which implies the low attribute leakage risk of the synthetic data generated by `PromptEHR`. Although the FPR becomes smaller than Control when $\delta > 0.8$, the TPR of `PromptEHR` is approaching zero after that. That means, being conservative for `PromptEHR` avoids inferring some wrong attributes but loses the ability to specify the right attributes at the same time. In a nutshell, the synthetic data by `PromptEHR` has a low risk to leak the attribute information.

4.5 Q3. Synthetic EHRs Utility

We aim to measure the utility of synthetic data when we develop predictive models on top of them. We compare LSTM models on $\mathcal{D}_S$ and $\mathcal{D}_1$ with multilabel prediction for diagnosis events similar to the setting in (Choi et al., 2016b). In particular, we design two experiments: (1) train LSTM on fully synthetic data and compare its performance with the one trained on real data; (2) train LSTM on a mixture of synthetic data and real data where the synthetic data is regarded as data augmentation.



(a) Perplexity (`lpl`).

(b) Perplexity (`mpl`).

Figure 8: Black solid lines show the spatial and temporal perplexities of `PromptEHR` with regard to varying input training record sizes. Red dotted lines show the `lpl` and `mpl` of baseline LSTM+MLP trained on all training records (∼40k).

Fully synthetic data. We test the LSTM performance on 5k, 10k, 30k, and 50k synthetic patient records. For comparison, the model performance on 5k and 10k real records are also tested. Results are shown in Fig. 5. For recall@10 in Fig. 5a, we can observe that though 10k synthetic records are not comparable to 5k real records, 30k synthetic records can reach a better performance than 10k real records. On the other hand, for recall@20 in Fig. 5b, we surprisingly find that 5k synthetic records achieve the same performance as the 5k real records. With more synthetic records involved, the 50k synthetic records-based LSTM outperforms its counterpart on 10k real records at last. This experiment demonstrates that synthetic EHRs by `PromptEHR` are sufficient to support healthcare applications. It is expected to achieve comparable performance by synthetic data as the real data.

Hybrid synthetc-real data. In Fig. 6, we randomly sample 10k real data from $\mathcal{D}_1$ and combine them with different sizes of synthetic data from $\mathcal{D}_S$. We find that the model trained on the augmented hybrid data has obvious advantages over its counterpart on the real data. With more synthetic records involved, the model gains better performance. This demonstrates the utility of synthetic data used as augmentation in low-resource cases. Besides, from Fig. 6 we identify this hybrid data is still inferior to the model trained on all real records. So we are curious about how many synthetic and real data we need to *outperform* this seemingly performance upper bound. In other words, can we beat the real data with the synthetic data?

We conduct the next experiment where 30k real data is combined with synthetic data. Note that we have around 40k real training records in total. Results are shown in Fig. 7. It can be seen that 50k synthetic records plus 30k real records train better models than on all the real data.

4.6 Q4. Quality w.r.t. Training Size

In practice, the original data source to be shared might be in limited size, which elicits a question on how much the generation quality of `PromptEHR` is influenced by the size of the training cohort. To answer this question, we sampled 5k, 10k, and 20k patient records from the training set and testify the perplexity of the learned `PromptEHR`. Results are illustrated by Fig. 8. We plot the performance of the baseline LSTM+MLP method trained on all real training records (∼40k) in red dotted lines for comparison. It shows that `PromptEHR` trained on 5k training records has worse generation quality than the baseline. When additional 5k records are involved, `PromptEHR` not only outperforms the LSTM baseline but also all other baselines reported in Table 2, which demonstrates that `PromptEHR` is amenable to low resources and superior than the baselines.

4.7 Case Study

We demonstrate two use cases of `PromptEHR`: *generating from scratch* (Table 3) and *generating by completion* (Table 4). While previous works handle the former, only `PromptEHR` handles the completion setting because it makes flexible conditional generation based on either patient features or previous events. In Table 4, our model begins from all diagnosis of one patient and then generates labtests via cross-modal imputation. Then, we randomly sample one procedure and let the model impute all the remaining procedures based on diagnosis and the labtests. Iteratively applying this strategy yields diverse and realistic EHRs via conditional generation. We provide explanations of the two synthetic records in Appendix §A.

5 Conclusion

In this paper, we study how to leverage real EHRs to train a prompt learning based generative language model for synthetic EHRs generation, namely `PromptEHR`. Unlike previous EHRs generation methods, `PromptEHR` is able to learn from and generate heterogeneous EHRs. To evaluate its performance, we draw the idea of perplexity from the text generation literature and propose two perplexity measures: spatial and temporal perplexity. Experiments on MIMIC-III data demonstrates the quality of generated EHRs are better than the baselines. The synthetic data provides both utility and privacy for downstream healthcare applications.

Limitations

This work seeks to generate synthetic records hence avoid sharing sensitive personal electronic healthcare records for the development of machine learning models. In our experiments, we find the generated synthetic records by PromptEHR are invulnerable to two adversaries: membership inference and attribute inference. However, there is still possibility that there exists some more advanced attacking methods which can take the advantage of synthetic records. Obviously we cannot exhaust all adversaries for empirical privacy evaluation. In this viewpoint, it is promising to investigate theoretic-guaranteed EHRs generation approach. For instance, we may draw the idea of differential privacy to enhance the current method to provide a complete privacy protection.

References

Ali Amin-Nejad, Julia Ive, and Sumithra Velupillai. 2020. Exploring transformer text generation for medical dataset augmentation. In *Language Resources and Evaluation Conference*, pages 4699–4708.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223. PMLR.

Mrinal Kanti Baowaly, Chia-Ching Lin, Chao-Lin Liu, and Kuan-Ta Chen. 2019. Synthesizing electronic health records using improved generative adversarial networks. *Journal of the American Medical Informatics Association*, 26(3):228–241.

Siddharth Biswal, Soumya Ghosh, Jon Duke, Bradley Malin, Walter Stewart, and Jimeng Sun. 2020. EVA: Generating longitudinal electronic health records using conditional variational autoencoders. *arXiv preprint arXiv:2012.10020*.

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):1–28.

Kyunghyun Cho, B van Merrienboer, Caglar Gulcehre, F Bougares, H Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*.

Edward Choi, Mohammad Taha Bahadori, Joshua A Kulas, Andy Schuetz, Walter F Stewart, and Jimeng Sun. 2016a. RETAIN: An interpretable predictive model for healthcare using reverse time attention mechanism. In *International Conference on Neural Information Processing Systems*, pages 3512–3520.

Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. 2016b. Doctor AI: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*, pages 301–318. PMLR.

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.

Limeng Cui, Siddharth Biswal, Lucas M Glass, Greg Lever, Jimeng Sun, and Cao Xiao. 2020. CONAN: Complementary pattern augmentation for rare disease detection. In *AAAI Conference on Artificial Intelligence*, volume 34, pages 614–621.

Khaled El Emam, Elizabeth Jonker, Luk Arbuckle, and Bradley Malin. 2011. A systematic review of re-identification attacks on health data. *PloS one*, 6(12):e28071.

Khaled El Emam, Sam Rodgers, and Bradley Malin. 2015. Anonymising and sharing individual patient data. *BMJ: British Medical Journal*, 350:h1139.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Annual Meeting of the Association for Computational Linguistics*, pages 889–898.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27.

Jiaqi Guan, Runzhe Li, Sheng Yu, and Xuegong Zhang. 2018. Generation of synthetic electronic medical record text. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 374–380. IEEE Computer Society.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. 2017. Improved training of Wasserstein GANs. In *International Conference on Neural Information Processing Systems*, pages 5769–5779.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1):1–9.

Rina Kagawa, Yukino Baba, and Hideo Tsurushima. 2021. A practical and universal framework for generating publicly available medical notes of authentic quality via the power of crowds. In *IEEE International Conference on Big Data*, pages 3534–3543. IEEE.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Dongha Lee, Hwanjo Yu, Xiaoqian Jiang, Deevakar Rogith, Meghana Gudala, Mubeen Tejani, Qiuchen Zhang, and Li Xiong. 2020. Generating sequential electronic health records using dual adversarial autoencoder. *Journal of the American Medical Informatics Association*, 27(9):1411–1419.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Annual Meeting of the Association for Computational Linguistics*, pages 4582–4597.

Claudia Alessandra Libbi, Jan Trienes, Dolf Trieschnigg, and Christin Seifert. 2021. Generating synthetic training data for supervised de-identification of electronic health records. *Future Internet*, 13(5):136.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021b. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*.

Joseph S Lombardo and Linda J Moniz. 2008. A method for generation and distribution of synthetic medical record data for evaluation of disease-monitoring systems. *Johns Hopkins APL Technical Digest*, 27(4):356.

Scott McLachlan, Kudakwashe Dube, and Thomas Gallagher. 2016. Using the caremap with health incidents statistics for generating the realistic synthetic electronic healthcare record. In *IEEE International Conference on Healthcare Informatics*, pages 439–448. IEEE.

Graham Neubig. 2017. Neural machine translation and sequence-to-sequence models: A tutorial. *arXiv preprint arXiv:1703.01619*.

Jing Qian, Li Dong, Yelong Shen, Furu Wei, and Weizhu Chen. 2022. Controllable natural language generation with contrastive prefixes. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2912–2924.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *Technical Report*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Divya Saxena and Jiannong Cao. 2021. Generative adversarial networks (gans) challenges, solutions, and future directions. *ACM Computing Surveys (CSUR)*, 54(3):1–42.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, pages 3–18. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Zifeng Wang, Rui Wen, Xi Chen, Shilei Cao, Shao-Lun Huang, Buyue Qian, and Yefeng Zheng. 2021a. Online disease diagnosis with inductive heterogeneous graph convolutional networks. In *Proceedings of the Web Conference 2021*, pages 3349–3358.

Zifeng Wang, Yifan Yang, Rui Wen, Xi Chen, Shao-Lun Huang, and Yefeng Zheng. 2021b. Lifelong learning based disease diagnosis on clinical notes. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 213–224. Springer.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling tabular data using conditional gan. *Advances in Neural Information Processing Systems*, 32.

Dian Yu, Zhou Yu, and Kenji Sagae. 2021. Attribute alignment: Controlling text generation from pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2251–2268.

Ziqi Zhang, Chao Yan, Thomas A Lasko, Jimeng Sun, and Bradley A Malin. 2021. SynTEG: A framework for temporal structured electronic health data simulation. *Journal of the American Medical Informatics Association*, 28(3):596–604.

Ziqi Zhang, Chao Yan, Diego A Mesa, Jimeng Sun, and Bradley A Malin. 2020. Ensuring electronic medical record simulation through better training, modeling, and evaluation. *Journal of the American Medical Informatics Association*, 27(1):99–108.

A Case Study

The first case was generated from scratch (Table 3), it describes a patient who goes into ICU because of a cesarean. During the operation, a test of Hematocrit should be conducted to ensure blood loss of the patient within the safe range. In the second visit, the patient suffers from a bacteria infection. The patient then receives a series of lab tests regarding the inflammation. And spinal tap is performed to help cure serious infections. Antibiotic drugs, e.g., Ampicillin Sodium and Gentamicin, are used to cure the patient. It can be seen that the generated events all center around the same topic (liveborn) and the longitudinal and cross-modal connections are coherent.

The second case was generated based on a real patient EHR by leveraging flexible imputation functions of PromptEHR (Table 4). The model scans through the record in time order. For each modality in a visit, we randomly choose to keep all events, remove all events, or remove a part at random. The imputed events are marked red. For example, in visit-1, the model takes the diagnosis codes with prompts as inputs and generates the lab tests. Then, the generated lab tests are involved in the input with prompts. In addition, the procedure 'Enteral infusion of nutrition' is also kept in the inputs. The model then generates the remaining procedures in this visit. This process repeats until reaches visit-6 where the real EHR ends.

In general, the events in the second case are coherent under the topic of pneumonia and heart failure. The patient is diagnosed as suffering from pneumonia due to bacteria with many complications like a hemorrhage of gastrointestinal tract, heart failure, and pulmonary collapse. At the same time, procedures like the enteral infusion of nutrition, insertion/replacement of endotracheal tube, and temporary tracheostomy are all included to maintain the patient's life regarding his/her nutrition and breath. Besides this visit, the remaining synthetic visits are also reasonable: he/she gets diagnoses regarding heart failure, respiratory diseases, stomach disorders, etc., which all correspond to relevant issues appearing in the first visit. These two cases offer an intuitive demonstration of the effectiveness of PromptEHR in generating realistic EHRs, especially when we take the advantage of multiple imputation functions to generate rather realistic EHRs based on real EHRs, which was hardly mentioned in previous works.

Table 3: A synthetic patient generated by PromptEHR from scratch. *ICD_abc* indicates the first three digits represented by ICD code of the event.

Visit-1	**Diagnosis**: Liveborn **Labtest**: Hematocrit **Procedure**: Prophylactic vaccination
Visit-2	**Diagnosis**: Streptococcus infection, Extreme immaturity, Perinatal infection, Neonatal jaundice, Liveborn **Labtest**: Anion Gap, Bands, Base Excess, Bilirubin, Total, Chloride, Eosinophils, Hematocrit, Hemoglobin, Lymphocytes, MCH, MCHC, MCV, Monocytes, Platelet Count, Potassium, Red Blood Cells, Sodium, pCO2, pH, pO2 **Drug**: Ampicillin Sodium, Heparin Sodium (Preservative Free), NEO*IV*Gentamicin, NEO*PO*Ferrous Sulfate Elixir, Send 500mg Vial, Syringe (Neonatal) *D5W* **Procedure**: Biopsy of spinal cord

Table 4: A synthetic patient generated by PromptEHR based on a real patient record. The imputed events are marked yellow. For demonstration, we cut the events after the fifth for each visit due to the space limit.

Visit-1	**Diagnosis**: Pneumonia, Hematemesis, Heart failure, Emphysema **Labtest**: Leukocytes , Urea Nitrogen , Calcium , Ketone **Procedure**: Enteral infusion of nutrition, Insertion of airway , Replace tracheostomy tube , Temporary tracheostomy
Visit-2	**Diagnosis**: Heart failure , Respiratory conditions , Tracheostomy status , Stomach disorder **Labtest**: Urine Appearance, Yeast, Platelet Count , Calculated Total CO2 **Procedure**: Biopsy of bronchus, Replace gastrostomy tube, Invasive mechanical ventilation , Infusion of nesiritide
Visit-3	**Diagnosis**: Pneumonia, Mechanical complication, Pulmonary manifestations , Disorders of urinary tract **Labtest**: INR(PT), Epithelial Cells, RBC, Urine Appearance **Procedure**: Insertion of airway, Enterostomy , Lysis of peritoneal adhesions , Lung biopsy
Visit-4	**Diagnosis**: Mechanical complication, Hodgkin's paragranuloma, Pressure ulcer , Heart failure **Labtest**: Urine Color , Urobilinogen , Bands , Urea Nitrogen **Procedure**: Infusion of nesiritide , Endoscopy of small intestine , Gastrostomy , Replace tracheostomy tube
Visit-5	**Diagnosis**: Urethra disorder, Attention to tracheostomy/gastrostomy, Pneumonia , Heart failure **Labtest**: MCH, Bacteria , Lymphocytes , Calculated Total CO2 **Drug**: Fluticasone Propionate 110mcg, SW , Bisacodyl , Iso-Osmotic Dextrose **Procedure**: Replace tracheostomy tube , Heart cardiac catheterization , Enteral infusion of nutrition
Visit-6	**Diagnosis**: Pneumonia, Heart failure, Endomyocardial fibrosis , Mechanical complication **Labtest**: pH, Epithelial Cells, WBC , Protein **Drug**: Neutra-Phos, Mirtazapine , Fluconazole , SW **Procedure**: Invasive mechanical ventilation, Airway infusion , Monitoring of cardiac output , Lung biospy