GenHPF: General Healthcare Predictive Framework for Multi-Task Multi-Source Learning

Kyunghoon Hur, Jungwoo Oh, Junu Kim, Jiyoun Kim, Min Jae Lee, Eunbyeol Cho, Seong-Eun Moon, Young-Hak Kim, Louis Atallah, and Edward Choi

***Abstract*—Despite the remarkable progress in the development of predictive models for healthcare, applying these algorithms on a large scale has been challenging. Algorithms trained on a particular task, based on specific data formats available in a set of medical records, tend to not generalize well to other tasks or databases in which the data fields may differ. To address this challenge, we propose General Healthcare Predictive Framework (GenHPF), which is applicable to any EHR with minimal preprocessing for multiple prediction tasks. GenHPF resolves heterogeneity in medical codes and schemas by converting EHRs into a hierarchical textual representation while incorporating as many features as possible. To evaluate the efficacy of GenHPF, we conduct multi-task learning experiments with single-source and multi-source settings, on three publicly available EHR datasets with different schemas for 12 clinically meaningful prediction tasks. Our framework significantly outperforms baseline models that utilize domain knowledge in multi-source learning, improving average AUROC by 1.2%P in pooled learning and 2.6%P in transfer learning while also showing comparable results when trained on a single EHR dataset. Furthermore, we demonstrate that self-supervised pretraining using multi-source datasets is effective when combined with GenHPF, resulting in a 0.6%P AUROC improvement compared to models without pretraining. By eliminating the need for preprocessing and feature engineering, we believe that this work offers a solid framework for multi-task and multi-source learning that can be leveraged to speed up the scaling and usage of predictive algorithms in healthcare.**

***Index Terms*—Electronic health records, heterogeneity, multi-source learning, multi-task learning, natural language process.**

Manuscript received 16 June 2023; revised 8 September 2023; accepted 13 October 2023. Date of publication 27 October 2023; date of current version 5 January 2024. This work was supported in part by the KAIST-NAVER Hyper-Creative AI Center, Institute of Information and Communications Technology Planning and Evaluation under Grant 2019-0-00075, in part by the National Research Foundation of Korea under Grant NRF-2020H1D3A2A03100945, in part by Korea Medical Device Development Fund under Grants 1711138160 and KMDF_PR_20200901_0097, and in part by Korea Health Industry Development Institute under Grant HR21C0198, funded by the Korea Government (MSIT, MOTIE, MOHW, MFDS). *(Corresponding author: Edward Choi.)*

Kyunghoon Hur, Jungwoo Oh, Junu Kim, Jiyoun Kim, Min Jae Lee, Eunbyeol Cho, and Edward Choi are with the Kim Jaechul Graduate School of AI, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea (e-mail: pacesun@kaist.ac.kr; ojw0123@kaist.ac.kr; kjune0322@kaist.ac.kr; jiyoun.kim@kaist.ac.kr; mjbooo@kaist.ac.kr; eunbyeol.cho@kaist.ac.kr; edwardchoi@kaist.ac.kr).

Seong-Eun Moon is with the Naver AI Lab, Seongnam-si 13561, South Korea (e-mail: seongeun.moon@navercorp.com).

Young-Hak Kim is with the Asan Medical Center, University of Ulsan College of Medicine, Songpa-gu 05505, South Korea (e-mail: mdyhkim@amc.seoul.kr).

Louis Atallah is with the Philips Research, Cambridge, MA 02141 USA (e-mail: louis.atallah@philips.com).

Our code implementation is available on Github. https://github.com/hoon9405/GenHPF

Digital Object Identifier 10.1109/JBHI.2023.3327951

I. Introduction

PATIENT medical records which are regularly accumulated in the form of Electronic Health Records (EHR) have opened up new opportunities for data-driven models, which can improve the quality of patient care. With the rapid adoption of artificial intelligence (AI) in healthcare, healthcare providers continue to develop models for different applications such as predicting patient outcomes [1], [2], [3], optimizing effective hospital operations [4], [5] and diagnosing diseases [6], [7], [8].

Until now, traditional model development methods have been constrained by their reliance on task-specific feature engineering, wherein preprocessing techniques are predominantly tailored for individual tasks or applications. For instance, predictive modeling tasks for patient care or benchmarking, and quality improvement require this approach. Consequently, each health system or research institute is compelled to employ its own data experts to meticulously preprocess medical records to suit specific tasks. This process can be time-consuming and expensive, ultimately restricting the range of potential applications [9].

Furthermore, this problem is exacerbated by the increasing number of tasks that require excessive overheads for the hospitals to develop and the managing of each task-specific model. Moreover, the increasing number of tasks significantly burdens hospitals in terms of developing and managing task-specific models. For example, clinicians may need to simultaneously perform various prediction tasks, such as mortality and readmission, for the same patient. To address this challenge, a comprehensive framework is required that can be applied to multiple tasks [10] with minimal preprocessing, thereby minimizing the need for a meticulous design of input features.

This problem contributes to the inequality in healthcare AI, as algorithms are developed and used by large (typically academic data centers) with access to large data and research capabilities. In reality, typical EHR datasets do not follow a single data format, particularly across geographies and multiple EMR providers. Each health system could store data according to its own needs, which consequently requires a level of manual harmonization.

Specifically, different EHR systems adopt different medical code standards (e.g., *ICD-9, ICD-10, raw text*), and use distinct database schemas to store patient records [11], [12], [13].[1] These discrepancies in medical codes and schemas prevent healthcare institutions from conducting multi-source learning, such as fine-tuning a model that has been previously trained on a different EHR dataset (i.e., *transfer learning*) or developing a unified model with data pooled from multiple hospitals (i.e., *pooled learning*).

In summary, the major challenges encountered by current healthcare prediction models are as follows: 1) models are specifically developed for each prediction task via feature engineering with task-specific domain knowledge, and 2) procuring a large amount of unified data is difficult, which is a critical problem for developing the aforementioned general-purpose multi-task prediction model. The main objective of this study is to propose a framework that addresses these two challenges.

Related work: Previous healthcare prediction models with EHR have been focused on increasing the prediction performance by utilizing domain knowledge and various architectures such as recurrent neural networks (RNN) [1], [14], convolutional neural networks [15], and transformer-based models [16], [17], [18], [19]. Although each study makes a distinct contribution, none address the two major aforementioned challenges.

Multi-Task Learning: MIMIC-Extract [20], for example, performs domain-knowledge-based feature engineering, such as grouping semantically similar concepts into a clinical taxonomy as data structures that are directly usable in common multi-task time-series prediction pipelines. Based on hand-crafted features, McDermott et al. [21] proposed a benchmark for ten healthcare predictive tasks (multi-task learning) and reported their prediction performances. Because of their specialized nature, these approaches are designed to work exclusively for specific datasets, making them inapplicable to multiple EHR datasets that may vary in diversity and heterogeneity.

An alternative approach, proposed by Rajkomar et al. [22], involves a framework that incorporates all features of the EHR, that is, all column values in all the EHR tables. This allows the same model to be used for four different tasks. However, since this approach uses Fast Healthcare Interoperability Resources (FHIR) [23], which is a form of Common Data Model (CDM), to manually standardize different EHR data into a uniform format, there is a significant overhead for multi-source learning. This process of standardizing EHR formats demands considerable domain knowledge and requires extensive manual efforts, making the integration of a large number of datasets into diverse formats impractical.

Resolving EHR Heterogeneity without manual efforts: To address the lack of scalability in previous works, AutoMap [24] conducts medical code mapping via self-supervised learning using a predefined medical ontology. This study aims to develop a solution to the current lack of a unified EHR system through a direct code-to-code mapping of two different medical institutions. However, since AutoMap requires standardized medical ontology, manual efforts is still necessary.

In another study, DescEmb [25] aimed to overcome the heterogeneity of medical codes by utilizing the clinical descriptions linked to each code, thereby partially enabling multi-source learning. Despite its text-based embedding to avoid the manual code mapping process, this approach still necessitates domain experts to conduct EHR system-specific preprocessing to select compatible and meaningful features from the EHRs. Overcoming schema heterogeneity across different institutions poses a challenge when selecting universally applicable features with consistent formats from multiple datasets. None of the aforementioned studies adequately address the dual challenges of utilizing multi-task models on heterogeneous EHRs.

Self-supervised pretraining in EHR: Self-supervised learning (SSL), which involves pretraining on large-scale unlabeled datasets and fine-tuning for prediction tasks, has demonstrated success in various applications [26], [27], [28] including predictive models based on EHR [29], [30], [31], [32], [33], [34]. Previous studies on SSL using EHR data have primarily focused on pretraining and fine-tuning models exclusively for identical EHR systems, limiting their applicability to other EHR systems. As the proposed framework resolves EHR heterogeneity, training it via SSL produces a general-purpose pretrained model that can be fine-tuned for any task in any EHR system.

This study makes three contributions. To address both challenges (task-specific model development process and EHR heterogeneity) simultaneously, we propose General Healthcare Predictive Framework (GenHPF) (Fig. 1), which is applicable to multiple patient record systems. GenHPF resolves heterogeneity in medical codes and schemas by converting medical records into a hierarchical textual representation while incorporating as many features as possible. This framework reflects the common data structure of medical records, allowing different structures to be utilized without code and schema harmonization processes.

Second, to demonstrate the efficacy of GenHPF empirically, we conduct extensive experiments using three publicly available EHR datasets with different schemas (MIMIC-III, eICU, MIMIC-IV) for the twelve clinically meaningful prediction tasks. Our framework achieves comparable or higher prediction performances on single-domain learning compared with other frameworks, while consistently outperforming all other frameworks in terms of pooled learning and transfer learning.

Lastly, we combine several SSL methods with GenHPF, demonstrating the best practices that provides benefits to GenHPF as a self-supervised pretraining method with unlabeled data. This will enable researchers and engineers in this field to use a pretrained GenHPF as a general-purpose foundation model for diverse prediction tasks, regardless of the EHR schema. Our findings provide insights for further research on the multi-source learning of EHR. Fig. 1 overviews the proposed framework.

II. Methodology

A. Structure of Electronic Health Records

This section describes and summarizes the EHR structure and notations used throughout this paper. In typical EHR data, each patient P can be represented as a sequence of medical events $[\mathcal{M}_1, \ldots, \mathcal{M}_N]$, where N is the total number of events throughout the entire patient visit history. The i-th medical event of a patient $\mathcal{M}_i$ can be expressed as a set of event-associated features $\{A_i^1, \ldots, A_i^{|\mathcal{M}_i|}\}$. Each feature A_i^k can be seen as a tuple of a feature name and its value $(n_i^k, v_i^k), n_i^k \in \mathcal{N}, v_i^k \in \mathcal{V}$, where $\mathcal{N}$ and $\mathcal{V}$ are each a set of unique feature names (e.g.,

[1] To unfold our tackling point conveniently, eICU is considered as EHR, which is originally collection across hospitals using different EHRs (EPIC, Cerner, etc.).

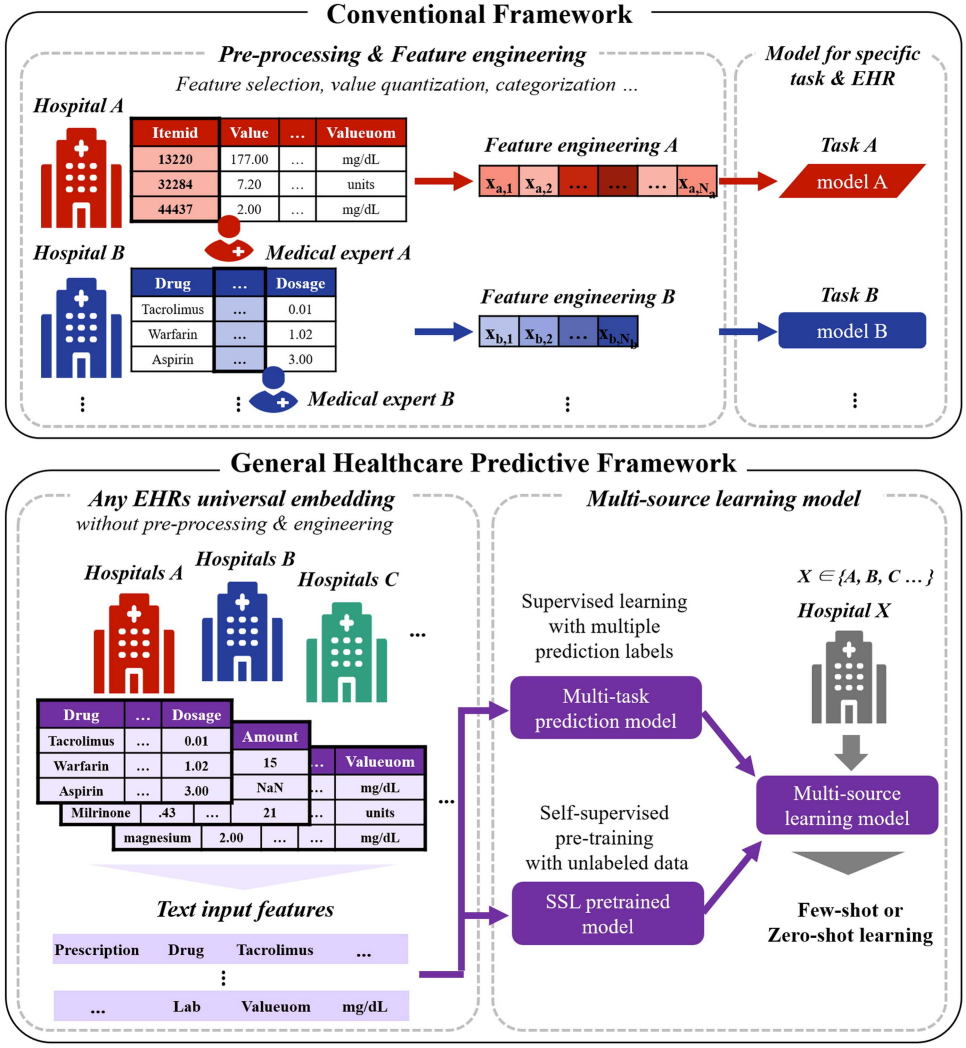

Fig. 1. Conventional approach for building predictive models uses domain-specific knowledge to preprocess data for each hospital (or health system) and task. In contrast, our proposed framework uses text input features, eliminating the need for preprocessing and feature engineering specific to each hospital. This allows us to train a unified model for two multi-source learning scenarios: 1) Conventional supervised learning for multi-task learning and 2) Self-supervised pretraining with unlabeled data. By employing transfer learning, our framework allows each trained model to conduct transfer learning in any hospital, irrespective of data format differences, thereby ensuring general adaptability across healthcare systems.

{"drug name", "drug dosage", ...,}) and feature values (e.g., {"vancomycin", "10.0", ...,}), respectively.

In addition, each medical event $\mathcal{M}_i$ has its corresponding event type $e_i \in \mathcal{E}$ which denotes the type of the event (e.g., $\mathcal{E} = \{$"*lab test*", "*prescription*", ...,$\}$). Lastly, since the recorded time is also provided with $\mathcal{M}_i$, we can measure the time interval t_i between $\mathcal{M}_i$ and $\mathcal{M}_{i+1}$.

B. General Healthcare Predictive Framework

In this section, we present GenHPF, a general framework for EHR-based prediction based on the following three principles, and describe how to implement each principle: 1) text-based embedding, 2) employing the entire features of EHR, and 3) medical event aggregation. Fig. 2 depicts the overall architecture.

Text-based embedding: A conventional EHR embedding method begins by assigning a unique embedding for each element in $\mathcal{V}$ via a linear map (i.e., lookup table) $f_{\mathcal{V}}$ [17], [21], [22], [35], [36], so that v_i^k can be converted to a vector $\mathbf{v}_i^k \in \mathbb{R}^{d_v}$, typically followed by pooling multiple feature values $(\mathbf{v}_i^1, \mathbf{v}_i^2, \ldots)$ to obtain $\mathbf{m}_i \in \mathbb{R}^{d_m}$, the embedding of $\mathcal{M}_i$.[2] This conventional embedding, however, usually requires a different $f_{\mathcal{V}}$ for each medical institution due to the $\mathcal{V}$ *heterogeneity*, namely each institution using different $\mathcal{V}$'s. For example, MIMIC-III [13], an open-source EHR data, uses the ICD-9 diagnosis codes for recording diagnostic information, while eICU [11], another open-source EHR data, uses in-house diagnosis codes. Therefore, the conventional embedding is not the most suitable foundation on which to build a general EHR framework.

DescEmb [25] proposed to resolve this problem by suggesting a text-based embedding, where hospital-specific feature values are first converted to textual descriptions (e.g., "401.9" $\rightarrow$ "unspecified essential hypertension"), then a text encoder paired with a sub-word tokenizer is used to obtain $\mathbf{m}_i$ [37]. With

[2]Previous EHR embedding methods do not typically use the feature name n_i^k

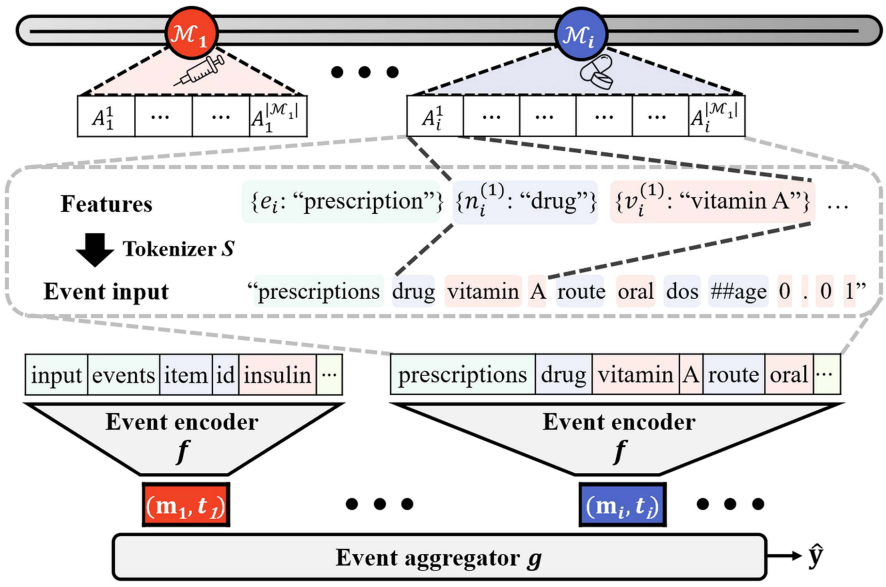

Fig. 2. Overview of GenHPF. On the top, a patient's medical events occur over time. Each medical event $\mathcal{M}_i$ consists of event-related features A_i^k, including feature names and their values. These features, prepended with event type e_i, are converted to corresponding descriptions and tokenized into a sequence of sub-words. Then, an event encoder f converts the sequence (i.e., event input) into an embedding $\mathbf{m}_i$, which is then passed to the event aggregator g, which then makes a prediction $\hat{\mathbf{y}}$.

this approach, the model can learn the language of the underlying medical text rather than memorize a unique embedding for each hospital-specific feature value, thereby overcoming the $\mathcal{V}$ *heterogeneity* as the same text encoder can be used for all institutions that use the same language. At this point, we adopted this code-agnostic embedding method and extended it by utilizing feature names as well as the feature values, which is $s(t(n) + t(v))$ as the event representation.

We extend the previous approach by applying the text-based embedding philosophy to event types e_i and feature names n_i^k, in addition to feature values v_i^k, as follows:

$$\mathbf{m}_i = f\left(S(e_i), S(n_i^1), S(v_i^1), \ldots, S(n_i^{|\mathcal{M}_i|}), S(v_i^{|\mathcal{M}_i|})\right)$$

where S is a sub-word tokenizer, and f is an event encoder that takes a sequence of sub-word tokens and returns $\mathbf{m}_i$. Note that f can be a pretrained language model as in DescEmb, or a randomly initialized transformer encoder, or even a single-layer RNN. Although f can be implemented with any sequence encoder (e.g., a pretrained language model as in DescEmb), we use 2-layer transformer in this work.

Employing the entire features of EHR: To develop a general predictive framework, in addition to the $\mathcal{V}$ *heterogeneity*, we must consider the *schema heterogeneity*, namely each medical institution using a different database schema. When developing a conventional predictive model, medical domain experts are typically involved to define $\mathcal{M}_i' \subset \mathcal{M}_i$, a subset of task-specific features among $\mathcal{M}_i$ according to each EHR system. This process must be carried out repeatedly whenever they encounter a different EHR schema. Moreover, in multi-source learning, medical domain experts must select and match compatible features between distinct EHR systems. For instance, in the *Lab* event of eICU, the feature named "labResult" should be paired with the "VALUENUM" feature in MIMIC-III's *LABEVENTS* event. Assessing database schemas of multiple sources and matching compatible features, although inevitable in a conventional approach, is time-consuming and prone to human errors.

Therefore, to leverage multiple heterogeneous EHR sources, features that share the same meaning must be matched. To avoid this costly procedure, our framework exploits the entire features of medical events, effectively resolving the schema heterogeneity. As described in II-B, the entire set of features in medical events is embedded into one unified embedding $\mathbf{m}_i$. Since this approach utilizes all features, feature selection is not required. Additionally, in multi-source learning, our framework is not constrained by the features that are present in each schema since both the name n_i^k and the value v_i^k of the feature are used. A formal comparison of the conventional approach, DescEmb [25] and our approach for obtaining $\mathbf{m}_i$ is provided below:

$$\textit{Conventionalapproach}:$$
$$\mathbf{m}_i = pool(\{f_{\mathcal{V}}(v_i^k) \mid A_i^k \in \mathcal{M}_i'\})$$
$$\textit{DescEmb}:$$
$$\mathbf{m}_i = f\left(\{S(v_i^k) \mid A_i^k \in \mathcal{M}_i'\}\right)$$
$$\textit{GenHPF}:$$
$$\mathbf{m}_i = f\left(S(e_i), \{S(n_i^k), S(v_i^k) \mid A_i^k \in \mathcal{M}_i\}\right)$$

where *pool* is typically implemented as a concatenation or summation of the elements. Note that GenHPF differs from previous approaches in that it is the only approach to exploit all available information in a medical event, including the event type, all event names, and all event values. Therefore, GenHPF provides a general solution applicable to any EHR system with a different schema, making it schema-agnostic, without requiring medical domain knowledge. DescEmb [25] still cannot resolve this since

it exploits only the feature value v_i^k. This approach does not take into account the need for the model to learn the semantics of column names, thereby necessitating only the selection of compatible features.

Medical event aggregation: To leverage the EHR structure characteristics, where P consists of a sequence of $\mathcal{M}_i$ and each $\mathcal{M}_i$ consists of a set of A_i^k, we design a hierarchical model consisting of the event encoder f, and the event aggregator g.

As each $\mathcal{M}_i$ is converted into $\mathbf{m}_i$ according to II-B, we can obtain $\mathbf{p} \in \mathbb{R}^{d_p}$, the vector representation of P as follows:

$$\mathbf{p} = g\left((\mathbf{m}_1, t_1), (\mathbf{m}_2, t_2), \ldots, (\mathbf{m}_N, t_N)\right)$$

where g is an embedding function that takes a sequence of event embeddings, and t is a timestamp which is applied as following [38], imposing the weight for attention according to the time interval between adjacent events. Note that g can be implemented with any sequence encoder, such as a Transformer encoder or a single-layer RNN. Then, feeding $\mathbf{p}$ through a softmax layer (sigmoid layer if binary prediction) will give us the final prediction $\hat{\mathbf{y}}$.

In addition, $\mathbf{p}$ can be obtained by employing a flattened model architecture rather than a hierarchical one, where sub-word tokens from all features of all medical events are passed to the sequence model h at the same time. We confirm that the hierarchical approach, which reflects the structure of EHR data, indeed outperforms the flattened approach.

C. Self-Supervised Pretraining

Building upon the premise that self-supervised pretraining may enhance downstream task performance, the proposed framework enbales multiple heterogeneous EHRs to be used during the self-supervised pretraining process. Our investigation focuses on determining the efficacy of various self-supervised pretraining approaches when applied to GenHPF. In this study, we test four well-known SSL methods as follows:

SimCLR [26]: We execute a two-step process of (1) EHR data augmentation and (2) contrastive pretraining inspired by SimCLR. For data augmentation, we create a pair of views per patient by halving the time-series data based on the number of events and randomly masking the tokens in the events at a fixed ratio. The contrastive pretraining objective is to maximize the similarity of the representation vectors created from two views of the same patient *(i.e., positive pair)* while minimizing the similarity of the vectors created from the views of different patients *(i.e., negative pair)* in accordance with the SimCLR settings [26].

Wav2Vec 2.0 [27]: We execute the Wav2Vec 2.0 [27] pretraining process, which consists of 1) feature encoder output quantization and 2) contrastive learning on mask-selected patient event timesteps. During the quantization stage, continuous latent vectors (i.e., event encoder outputs) are quantized via mapping the vectors to discrete entries of a trainable codebook. Gumbel softmax is used to map each latent vector to the codebook entries. During the second stage, a proportion of the latent vectors are randomly masked before being fed into the event aggregator. For each mask selected position, the overall pretraining objective is to maximize the similarity between the event representation vectors (i.e., event aggregator outputs) and their corresponding quantized vector, while minimizing the similarity with other quantized vectors. The loss terms are followed as defined in Wave2Vec 2.0. We use the event encoder as the feature encoder instead of the convolutional blocks used in the original study.

MLM and SpanMLM [28], [39]: For MLM pretraining, we randomly mask a fixed ratio of tokens among the whole patient event history, and the pretraining objective is to predict the masked tokens based on bidirectional attention. For SpanMLM pretraining, we apply event-level random masking, where all tokens included in the sampled events are masked, which is intended to learn the context of the EHR time-series event by learning the event itself rather than simply learning the partial random masked sub-word of the description. Note that both MLM and SpanMLM are based on predicting the raw text (i.e. tokens), which prevents us from using the hierarchical textual representation (Fig. 4). Therefore we use a flattened textual representation for these two methods; the Methods section describes this representation further.

III. Experimental Settings and Design

A. Datasets

We use three publicly available datasets; MIMIC-III [13], MIMIC-IV [12], and eICU [11]. The MIMIC-III database consists of clinical data of over 40,000 patients admitted to the intensive care units (ICU) at the Beth Israel Deaconess Medical Center. MIMIC-IV is an enhanced version of MIMIC-III that incorporates additional data sources, including admission date. The eICU consists of ICU records from multiple US-based hospitals, with 140,000 unique patients. All three datasets contain patient medical events including lab tests, prescriptions, and input events (e.g., drug injection), which are processed as inputs for the experiments. Each event is marked with a timestamp. We build patient cohorts of patients over the age of 18 years who remained in an ICU for over 24 hours. To ensure reliable experiments and analyses, we randomly split each dataset into training, validation, and test sets in an 8:1:1 ratio.

Minimal preprocessing applicable to any EHR is performed in three steps. First, we eliminate features whose values consisted only of integers. This approach ensures that all continuous-valued features (e.g., lab test results) and textual features (e.g., lab test names) are used, while omitting features such as the patient ID. Second, we split numeric values digit by digit and assign a unique token to each digit place, a method known as *digit place embedding* which was first introduced in DescEmb [25]. Subsequently, we tokenize all features and prepare them as text input features using bio-clinical-bert tokenizer [40]. Table I summarizes the general characteristics of the three datasets including the size and feature dimensions. The embedding method for each feature is either a categorized feature (code-based embedding) or is the text itself.

For the pretraining dataset, we prepare an unlabeled dataset, employing multiple ICUs without an observation window, and sampled medical events with a maximum length of 150, except for the test set of the downstream task. For medical sequences exceeding 150 events, we shift the starting point of sampling by 30 events, thereby altering the sample while maximizing data inclusion.

B. Prediction Tasks

To fairly evaluate our framework for varous healthcare predictive tasks, we utilize open-source prediction tasks that can

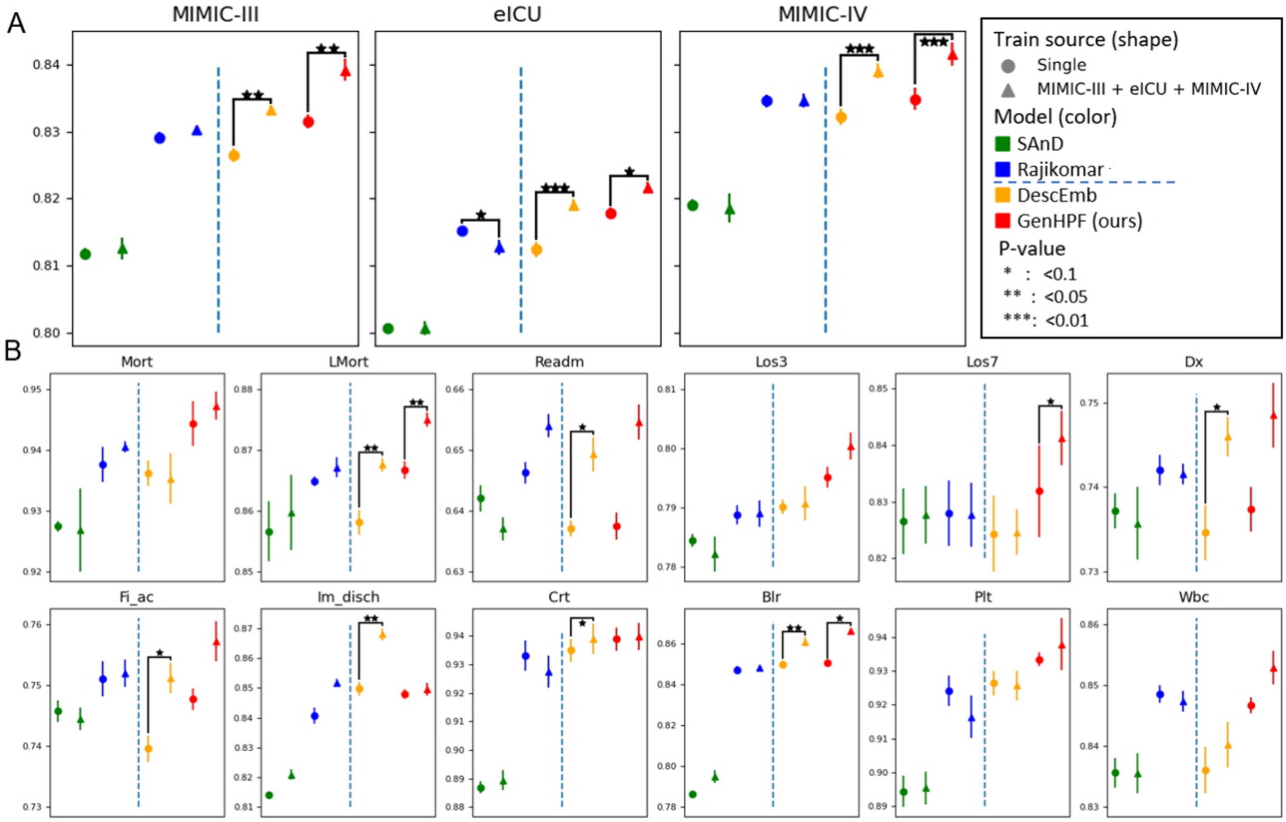

Fig. 3. Comparison of single domain learning and pooled learning prediction performances. (a) Results of the average AUROC on 12 prediction tasks. The data sources used for the evaluation are at the top of each graph. The y-axis indicates the AUROC. Each dot represents models (color) with source datasets used for training (shape) following the legends. Note that "Single" refers to the same data source as the evaluation dataset. The blue dashed line separates models into conventional embedding models (left- SAnD, Rajkomar) and text-based embedding models (right- DescEmb, GenHPF). Stars indicate the p-value of the t-test conducted to assess the significance between single-domain prediction and pooled learning. (b) Results of each prediction task using MIMIC-III as the source dataset.

TABLE I
CHARACTERISTICS OF DATASETS

Statitics	MIMIC-III		eICU		MIMIC-IV	
No. of Observations	38040		98904		65511	
No. of ICU stay	38040		98904		65511	
Mean No. of events per sample	102.5		48.5		88.7	
Feature selection	-	O	-	O	-	O
No. of Unique code	10434	6370	6302	5704	9565	5808
No. of Unique subwords text	3321	2793	2678	2451	3512	3112
Mean No. features per event	7.2	4.5	6.7	5.2	10	4.4
Mean length of subwords text per event	44.6	25.8	51	34.6	62.2	24.7

be applied in an ICU setting. We adopt eight prediction targets (Mort, LMort, Readm, Los3, Los7, Dx, Fi_ac, and Im_disch), as described by McDermott et al. [21]. Additionally, to demonstrate the efficacy of GenHPF in a broader range of tasks, we formulate four prediction targets for lab values, which serve as proxy indicators for sepsis or acute kidney injuries [41]. All tasks are based on ICU stays, and the performance is evaluated using the area under the area under the receiver operating characteristic curve (AUROC). Each task is defined as follows:

- *Mortality* (Mort) (binary): A sample is labeled positive for mortality if the discharge state was "expired" within a prediction window of 48 hours during the stay. In addition, for a longer-term prediction mortality prediction, we use death within 2 weeks (abbreviated as LMort).
- *Length-of-Stay* (LOS) (binary): The length of stay prediction for ICU stays can be categorized into two cases: determining whether a given stay lasted longer than 3 days (LOS3), and determining whether it lasted longer than 7 days (LOS7).
- *Readmission* (Readm) (binary): Given a single ICU stay, we consider a positive case of an ICU stays followed by another (readmission) during the same hospital stay.

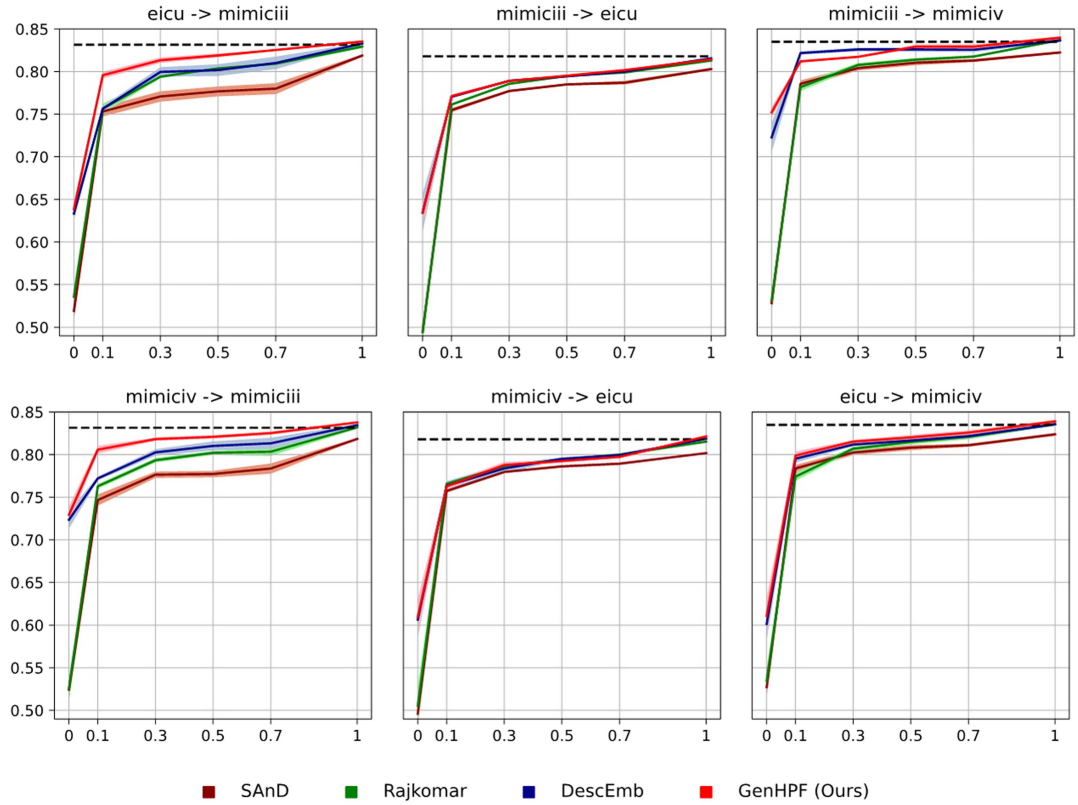

Fig. 4. Transfer learning results. The source data used for training and target data used for evaluation with zero-shot or few-shot learning are indicated at the top of each graph. The source dataset is on the left side of the arrow, and the target is on the right. The y-axis indicates the AUROC and the x-axis is the portion of the target dataset for zero-shot or few-shot learning. Shading around the lines indicates the standard error from five seed experiments. For comparison with single domain performances, single domain learning performances of GenHPF are marked with the dashed line.

- *Final Acuity* (Fi_ac) (multi-class): Predicts the patient's discharge location at the end of their hospital stay, including patient expiration.
- *Imminent Discharge* (Im_disch) (multi-class): Predicts whether the patient will be discharged within a prediction window of 48 h and if discharged, predicting the discharge destination.
- *Diagnosis* (Dx) (multi-label): Predicts all diagnosis (Dx) codes accumulated during an entire hospital stay. We group Dx codes into 18 Dx classes using Clinical Classification Software (CCS) for the ICD-9-CM criteria [healthcare2016hcup].
- *Lab values* (multi-class): Four distinct laboratory values[Creatinine (Crt), Bilirubin (Blr), Platelets (Plt))] are categorized into 5 classes, based on their corresponding ranges. These classes are derived using the thresholds employed to determine the Sequential Organ Failure Assessment (SOFA) scores. White blood cell (Wbc) is categorized into 3 classes.

Using medical event information from the initial 12 hours after ICU admission, we apply a 12-hour timegap across all tasks. timegap, which is designed to exclude any data close to the prediction time, is implemented to maintain the task challenges and prevent potential data leakage. Excluding any ICU stays shorter than 24 h allows for both a 12-h observation window and a 12-h gap. For diagnosis, we categorize diagnosis labels into 18 distinct classes based on the CCS ontology [42]. We use ICD9 for MIMIC-III, ICD10 for MIMIC-IV, and text format diagnostic labels for the eICU. We perform an additional mapping process for Fi_ac and Im_disch owing to the different labeling sources across datasets. For the lab value prediction tasks, we adopt the approach of Gyawali et al. [43], defining them from the SOFA score, which guides the severity of sepsis from specific lab values. Hence, each lab value is assigned as a categorical value based on its corresponding SOFA score. Statistics for prediction tasks are shown in Tables V, VI, VII.

C. Baselines and Implementation Details

Baselines: As there is no previous work, to our knowledge, that tackled exactly the same goal as ours, we modified well-known general-purpose EHR embedding frameworks. By comparing GenHPF with baselines, we systematically evaluate the components that can influence prediction performance in multi-source learning settings. We analyzed these frameworks based on two options: feature utilization (selective or utilizing all) and embedding method (code-based or text-based). In addition, all models were provided with both n_i^k and v_i^k for a fair comparison with GenHPF.

- *SAnD:* This uses the conventional embedding, selected features $\mathcal{M}_i'$, and the flattened architecture, similar in spirit to SAnD [17]. Note that feature embeddings from all medical events $[\mathcal{M}_1, \ldots, \mathcal{M}_N]$ are directly fed to the

sequence encoder h instead of being pooled to obtain individual $\mathbf{m}_i$.

- *Rajkomar:* This uses the conventional embedding, entire features $\mathcal{M}_i$, and the hierarchical approach, similar in spirit to [22] except the CDM standardization. Note that the feature embeddings from each $\mathcal{M}_i$ are fed to f to obtain individual $\mathbf{m}_i$, which is then fed to g.
- *DescEmb:* This uses the text-based embedding, selected features $\mathcal{M}'_i$, and the hierarchical approach, similar in spirit to DescEmb [25].
- *AutoMap:* This uses the same embedding method and features as Rajikomar [22]. It trains $\mathcal{M}_i$ by automatically mapping medical codes using ontology-level alignment with an unsupervised learning method.
- *Muse:* This uses the same embedding method and features as Rajikomar [22]. It trains $\mathcal{M}_i$ using skip-grams and aligns the embedding space between bilingual dictionaries.

Model implementation: For a fair comparison, f and g were both implemented with a randomly initialized 2-layer Transformer encoder, and a 4-layer Transformer encoder, making all models equivalent in terms of the number of trainable parameters ($d_v = 128,\ d_m = 128,\ d_p = 128$).[3] Although all frameworks share the same sequence of medical events, the selection of features and the embedding approach employed can vary across each frameworks. The selected features $\mathcal{M}'_i$[4] followed by DescEmb [25].

To maintain the same input information for both hierarchical and flattened models, we limit the number of events per sample. Owing to computational resource constraints, the flattened models are limited to a maximum sequence length of 8192, and a correspondingly adjusted number of events were used as input for the hierarchical model, which includes the same events as the flattened model.

Training details: All experiments are conducted using five random seeds which are used to initialize the model parameters and to split the dataset. Their performance is evaluated based on the area under the receiver operating characteristics (AUROC) averaged over twelve tasks. We conduct all experiments in a multi-task learning setting, as our main interest is to develop a single model that performs multiple tasks using multiple EHR datasets simultaneously. For multi-source learning, we train the combined dataset and validate each individual dataset separately. Early stopping is enforced according to the validation AUROC for each dataset, and the best model is saved per dataset. Subsequently, each saved model is used to test the corresponding dataset.

Hyperparameters: We explored various hyperparameters to determine the optimal for each framework. However, we found that the impact of these hyperparameters on the results was not significant. Consequently, we use a unified set of hyperparameters for all cases, thereby simplifying the experiment while maintaining the performance for each model. The final hyperparameters are a dropout of 0.3, a batch size of 64, and a learning rate of 1e-4. For pretraining, we apply token masking with the same fixed ratio to SimCLR, MLM, and SpanMLM, in which 80% of the randomly chosen token positions are replaced with the [MASK] token, 10% of the positions are replaced with a random token, and the remaining 10% of the positions are unmodified. We apply Wav2Vec settings with 2 codebooks, 320 entries per codebook, a masking ratio of 65%, and a feature gradient multiplication of 0.1 which slows down the event encoder gradient update. For the codebook diversity loss weight, we use 0.1, 0.3, 0.1, 0.5 for MIMIC-III, eICU, MIMIC-IV, and the pooled domain, respectively.

[3]Note that we modified each baseline from the original frameworks for a fair comparison (e.g. transformer architecture [16] which is a state-of-the-art model, is used instead of RNN.)

[4]For example, from the prescription event, we chose essential features such as drug name, drug volume, and unit of measurement among all available features.

D. Experimental Design

To assess the efficacy of GenHPF in various aspects, we developed a series of prediction tasks across four distinct scenarios: 1) single-domain learning, 2) pooled learning, 3) transfer learning, and 4) self-supervised learning. For pooled learning and transfer learning, we follow the settings from DescEmb [25]. For single-domain learning, models are trained and tested on a single dataset. This part tests GenHPF for single-domain learning although its primary aim is that of multi-source learning. In pooled learning, it is crucial to utilize data collected from multiple EHR systems by leveraging the wealth of EHR data for prediction tasks. Each framework simultaneously is trained on all three datasets, and evaluated separately on each dataset. We compare the performance of single-domain learning and pooled learning to show that training on multiple datasets enhances predictive performance compared with models trained on a single dataset.

Next, in transfer learning, we aim to show that GenHPF can be beneficial when trained on a specific dataset and directly tested on other datasets (zero-shot learning) or when further trained on limited data (few-shot learning). In practice, a single deep-learning model is typically trained on a large-scale hospital dataset and subsequently transferred to individual institutions, which could enable small hospitals to benefit from models trained on a large scale. Apart from acquiring large and representative datasets, this also entails ensuring compatibility between code and data schemas across different EHR systems, akin to what is necessary in pooled learning. In this scenario, each model is first trained on a source dataset and then directly evaluated on a sample from the same dataset (i.e., zero-shot) or further trained (i.e., fine-tune) on a target dataset.

Finally, we investigate which SSL method with unlabeled data exhibits a performance improvement when fine-tuning the pretrained model on the prediction task. To demonstrate the benefit of our approach, we compare three models: 1) a randomly initialized model trained on a single dataset; 2) a pretrained model and fine-tuned on a single dataset; 3) a pretrained model on the multi-source (pooled) dataset and fine-tuned on a single dataset. Additionally, we assess the impact of pretraining on different fine-tuned data size settings, namely sample data, and full data, assuming that a pretrained model can be fine-tuned on a smaller hospital or a similar-sized hospital.

IV. Results

A. Single-Domain Learning

Fig. 3 shows the single-domain learning results. GenHPF shows comparable or higher prediction performances, on average, across the 12 tasks than other frameworks using domain knowledge (+0.8%P AUROC on average against all frameworks on all three datasets, Fig. 3(a) circle marks) Appendix A provides

TABLE II
SELF-SUPERVISED PRE-TRAINING RESULTS

		Structure	Hierarchical			Flatten		
FT Source	FT data size	PT Source	Rand.Init	SimCLR	Wav2vec	Rand.Init	MLM	SpanMLM
MIMIC-III	Sample data (10%)	single	0.721	0.752***	0.733*	0.711	0.716	0.709
		multi-source		**0.769***(0.006)**	0.74**		0.709	0.701*
	Sample data (30%)	single	0.783	0.799**	0.779	0.767	0.773*	0.757**
		multi-source		**0.805***(0.008)**	0.781		0.77	0.755**
	Full Data	single	0.831	0.83	0.832	0.804	0.805	0.799*
		multi-source		**0.840**(0.011)**	0.832		0.812**	0.801
eICU	Sample data (10%)	single	0.721	0.768***	0.737**	0.708	0.72*	0.695**
		multi-source		**0.771***(0.008)**	0.734**		0.725**	0.689***
	Sample data (30%)	single	0.789	0.793	0.785	0.773	0.77	0.757**
		multi-source		**0.801**(0.049)**	0.79		0.78*	0.768
	Full Data	single	0.817	0.818	0.818	0.802	0.801	0.797
		multi-source		**0.82(0.118)**	0.818		0.805	0.796*
MIMIC-IV	Sample data (10%)	single	0.707	0.729***	0.717**	0.698	0.711**	0.677***
		multi-source		**0.735***(0.008)**	0.716**		0.709**	0.688*
	Sample data (30%)	single	0.761	0.776**	0.765	0.753	0.751	0.744*
		multi-source		**0.782***(0.009)**	0.764		0.757*	0.751
	Full Data	single	0.834	0.837	0.834	0.814	0.815	0.808*
		multi-source		**0.842**(0.014)**	0.835		0.817	0.809*

p-value *:<0.1, **: <0.05, ***: <0.01

The t-test significance values for the bolded values.

the comparison results of GenHPF with a baseline involving more feature engineering.

B. Pooled Learning

The results reveal that GenHPF exhibits a significant improvement in pooled learning when trained simultaneously on all three datasets, outperforming all other frameworks (+1.2%P, Fig. 3(a) triangles). This highlights the advantages of GenHPF which utilizes the textual representations of all features. Compared with single-domain learning results, text-based embedding models (DescEmb and GenHPF) consistently demonstrate higher performances when trained on pooled datasets from all three sources. In contrast, conventional embedding models (SAnD and Rajkomar) show decreased or unchanged performances for pooled learning. In addition, for text-based embedding models, GenHPF outperforms DescEmb in most cases when all three data sources are pooled together.

C. Transfer Learning

Fig. 4 presents the transfer learning results. To evaluate how the performance of the frameworks varies with the target dataset size, we use different proportions of the target dataset: x = 0.0 indicates zero-shot learning, x = 0.5 means fine-tuning with half of the target dataset, and x = 1.0 is for fine-tuning with the entire target dataset. For zero-shot learning, the text-based embedding methods (DescEmb and GenHPF) consistently outperform the code-based embedding methods (SAnD and Rajikomar) across all source and target pairs.

GenHPF demonstrate predominantly higher performance than the other models in most cases (+2.6%P, red line over other lines). As the sample size of the target dataset decreases, the strength of GenHPF becomes more apparent (+12.5%P, performance at x = 0.0). In further fine-tuning on the full dataset (marked with 1 on the x-axis), the code-based embedding models perform worse than GenHPF with single-domain learning performance (dotted line) in most cases. In contrast, GenHPF exhibits comparable or higher performance than single-domain learning, except when the model is trained on MIMIC-III and transferred to the eICU. Next, we introduce two additional baselines capable of automatically map different code systems between two EHR datasets using unsupervised learning. GenHPF exhibits a higher performance against unsupervised learning methods for code mapping, as shown in Table IV.

D. Self-Supervised Pretraining

Table II presents the results. Pretraining sources (PT Srouce) are in two settings, single(same as the fine-tune dataset) and multi-source (MIMIC-III+eICU+MIMIC-IV). Fine-tune(FT) data size are varied with the data sampled size (10%, 30%, full). $\star$ indicates p-value from the t-test between the randomly initialized model and pretrained model results. The highest performance for each fine-tune source, corresponding to the size of the fine-tune data, is highlighted in bold, and its p-value is indicated in parentheses.

The results show that GenHPF coupled with self-supervised pretraining methods (except SpanMLM) improves the prediction performance in most cases compared to models without pretraining. Among the pretraining methods, SimCLR consistently outperforms the others, exhibiting the highest prediction AUROC, for both the sample-data and full-data scenarios. In particular, SimCLR exhibits an average increase of 0.1%P and 0.6%P in the AUROCs for the single- and multi-source pretraining, respectively. The sample data results show that when the quantity of pretraining data exceeds that of the fine-tuning data to a larger extent, pretraining significantly affects the predictive downstream tasks.

V. DISCUSSION

In this work, we addressed the dual challenges of multi-task prediction models for heterogeneous EHRs by proposing and investigating GenHPF for single-domain learning, pooled learning, transfer learning, and self-supervised pretraining. The results show that GenHPF achieves comparable or higher performances without relying on medical domain knowledge and by simply using all features as textual descriptions.

In particular, for single domain learning, a comparison between GenHPF and Rajkomar suggests that assigning unique

embeddings to all feature names and values is unnecessary, since treating them as textual descriptions leads to a comparable performance. Moreover, a comparison between GenHPF and DescEmb implies that GenHPF can better capture the underlying semantics of distinct EHR sources than DescEmb utilizing all available information in a medical event. That is, applying medical domain knowledge to select a subset of meaningful features does not necessarily lead to a higher performance compared with simply using all possible features. Overall, the single-domain learning results show that GenHPF achieves comparable or higher performances, even without relying on medical-domain knowledge, by simply using all features as textual descriptions. The improved AUROC achieved without significant feature engineering made this evident.

In the pooled learning, both text-based embedding models (DescEmb and GenHPF) significantly improved the prediction performance compared with conventional code-based embedding models. This improvement results from the MIMIC and eICU datasets not sharing codes and from training conventional code-based embedding models on the pooled dataset expanding the number of required embeddings for each feature name and value, thereby preventing the models from leveraging larger amounts of training data. Conversely, text-based embedding models can take advantage of the extensive volume of various sources since the sub-words of medical descriptions are common, even among entirely dissimilar EHR systems. Furthermore, even within the text-based embedding models, GenHPF outperforms DescEmb in most cases, although DescEmb uses manually selected features from each dataset. This highlights the advantage of GenHPF because it does not rely on any domain knowledge but rather uses all features in a textual form regardless of the EHR schema used.

In transfer learning, we observe a pattern similar to that in pooled learning; text-based embedding models consistently outperform code-based embedding methods. GenHPF demonstrates performances similar to or better than those of DescEmb, except when transferring from MIMIC-III to MIMIC-IV with 10-30% fine-tuning. Through these experiments, we demonstrate that GenHPF effectively resolves two challenges (multi-task learning, multi-source learning). For multi-task learning, GenHPF outperforms models SAND and DescEmb, which employ feature selection by utilizing domain knowledge. Regarding multi-source learning, GenHPF demonstrated better performance than conventional embedding models such as Rajkomar and SAnD.

The self-supervised pretraining results show that SimCLR consistently outperforms the other methods. We conjecture that SimCLR's pretraining process effectively facilitates prediction in downstream tasks by learning patient-level representations, whereas the other pretraining methods focus on learning either token-level or event-level representations within the same patient. Furthermore, the performance improvement of GenHPF with multi-source pretraining provides insights into the necessity of pretraining on the pooled heterogeneous EHRs, which we believe is essential for large-scale EHR modeling.

Implementing GenHPF in a real-world hospital requires appropriate hardware resources, including GPUs connected to EHR database. Once operational, the framework minimally preprocesses patient data for various prediction tasks. A key advantage of GenHPF is that it can be integrated into any EHR system without requiring specific modifications, thereby significantly reducing both time and implementation costs. However, this approach to minimal preprocessing results in a larger input size, requiring higher computational requirements.

VI. Limitation

Although GenHPF demonstrated promising results, it still has limitations. First, since GenHPF utilizes as many features as possible from EHR events, computational constraints must be considered. Therefore, we used a subset of EHR events (lab tests, prescriptions, and input events) in this work. Better performance is expected if we exploit all EHR event types using more memory-efficient models [45], [46].

Second, as the current framework for multi-source learning relies on textual representation, it is limited to EHRs that share the same language. Lastly, we used only tabular data in the EHR; thus, future studies should consider incorporating additional modalities (e.g., radiographic images) into the framework.

VII. Conclusion

In conclusion, our study illustrates the potential of GenHPF for various learning scenarios, including single-domain, pooled, transfer learning, and self-supervised pretraining. The effectiveness of the framework without relying on medical domain knowledge and its ability to capture the underlying semantics of distinct EHR sources make it a promising approach for large-scale EHR modeling in the future. Furthermore, With the advent of large language models (LLMs) such as Chat-GPT, feeding text-based EHRs into an LLM via the GenHPF framework (with its ability to handle any EHR in text form) would allow for EHR predictions, either by fine-tuning the LLM or using the in-context learning technique. This would open up a wide set of applications that could reduce complications and improve patient care with less reliance on EHR schemas and feature engineering, such as predicting patient outcomes, intervention, and personalizing patient care.

Appendix A
Supplementary Results

A. Comparison of GenHPF With Benchmark [21]

To provide more credibility, we compare GenHPF with Benchmark [21], shown in table III. Statistical significance of the differences in the AUROC scores is reported and denoted by * (p-value $<$ 0.1), ** (p-value $<$ 0.05), and *** (p-value

TABLE III
Comparison With Benchmark Model (Only Lab Features)

Source	Benchmark	Rajikomar	GenHPF (ours)
MIMIC-III	0.779	**0.786**(0.031)**	0.784*
eICU	0.783	0.788*	**0.79**(0.024)**

The t-test significance values for the bolded values.

TABLE IV
Transfer Learning With Additional Baselines

Source ->Target		AutoMap	MUSE	Rajikomar	GenHPF
MIMIC-IV ->eICU	Zero-shot	0.473***	0.502***	0.505***	0.610
	Finetune	0.811*	0.812*	0.815	0.821
eICU ->MIMIC-IV	Zero-shot	0.531***	0.52***	0.535***	0.611
	Finetune	0.829**	0.831*	0.836	0.839
MIMIC-IV ->MIMIC-III	Zero-shot	0.509***	0.524***	0.525***	0.729
	Finetune	0.817***	0.826**	0.832*	0.838

TABLE V
STATISTICS FOR BINARY CLASSIFICATION TASKS

	MIMIC-III		eICU		MIMIC-IV	
Task/Class	0	1	0	1	0	1
mortality	98.3	1.7	98.4	1.6	98.4	1.6
long_term_mortality	91.6	8.4	92.8	7.2	92.3	7.7
los_3day	65.8	34.2	71.2	28.8	70.8	29.2
los_7day	87.9	12.1	91.2	8.8	90.9	9.1
readmission	94.5	5.5	90	10	92.7	7.3

< 0.01). The best performances for each dataset are in bold. While Benchmark offers an expert-designed, feature-engineered prediction pipeline, comparing it with GenHPF allows us to assess the effectiveness of our method, which operates without domain-specific knowledge. Benchmark originally used all tables, including lab tests and chart events. Due to the high computational demands from numerous chart events, we limited our comparison to the lab test table. This ensures a fair comparison, as both our method and Benchmark share only the lab test event. GenHPF generally exhibits a higher performance than that of Benchmark in most prediction tasks.

B. Comparison GenHPF With Unsupervised Learning Methods in Transfer Learning

AutoMap [24] and Muse [44] use the same model architecture as Rajikomar [22] but can leverage learned embedding through the unsupervised pretraining of code features between the source and target datasets. We use these baselines for fair a comparison when transferring code-based embedding models, giving pretrained embedding, not just randomly initialized. Results are shown in Table IV. The two unsupervised learning methods for code-mapping do not exhibit improvement over Rajkomar in the full dataset performance. This indicates that pretraining with code-mapping between two sources using different EHR code schemes does not yield a performance improvement, and the original paper did not conduct the experiments across different EHRs. However, GenHPF which utilizes text-based embedding outperforms the baselines (AutoMap, Muse, and Rajikomar) in both zero-shot learning and full dataset fine-tuning.

APPENDIX B
STATISTICS FOR PREDICTION TASKS

This section presents the statistics for the prediction tasks. All numbers represent the composition ratios as percentages.

Binary Classification Tasks Table V The tasks include predicting mortality, long-term mortality, los3, los7, and readmission. The values represent the percentage of total instances for each class in the corresponding dataset (MIMIC-III, eICU, MIMIC-IV).

Multi-class Classification Tasks Table VI The tasks include predicting the final acuity, imminent discharge, and several lab values (creatinine, bilirubin, platelets, and WBC). For laboratory values, 'Null' denotes ICU samples that involve dialysis. The loss is not computed for this null class during training phases. For the final acuity and imminent discharge, samples outside the predefined classes are marked as 'Null'.

TABLE VI
STATISTICS FOR MULTI-CLASS CLASSIFICATION TASKS

Task	Class	MIMIC-III	eICU	MIMIC-IV
final_acuity	Null	0.0	1.0	0.8
	0	51.9	58.5	51.5
	1	3.4	3.4	3.7
	2	7.2	5.0	6.0
	3	9.6	13.2	11.5
	4	12.4	4.6	7.3
	5	15.5	13.6	18.6
imminent_discharge	Null	0.0	0.1	0.5
	0	95.4	1.6	1.6
	1	0.2	5.5	2.9
	2	1.7	90.9	93.9
	3	0.5	1.6	0.5
	4	0.1	0.0	0.0
	5	0.1	0.1	0.1
creatinine	Null	15.7	25.1	14.8
	0	59.2	49.8	58.8
	1	16.1	15.1	16.4
	2	6.2	6.5	6.4
	3	1.6	1.9	1.9
	4	0.1	1.6	1.8
bilirubin	Null	78.5	71.4	74.0
	0	13.2	21.9	16.4
	1	2.8	3.3	3.4
	2	3.6	2.5	4.1
	3	1.0	0.5	1.1
	4	0.8	0.3	0.9
platelets	Null	11.9	25.2	11.7
	0	61.8	49.4	58.0
	1	16.9	15.9	19.1
	2	7.5	7.2	8.6
	3	1.7	1.6	2.2
	4	0.3	0.3	0.4
wbc	Null	12.3	24.7	11.8
	0	3.6	45.5	3.9
	1	53.5	26.6	55.0
	2	31.0	2.6	29.4

TABLE VII
STATISTICS FOR MULTI-LABEL CLASSIFICATION TASK(DX)

class	MIMIC-III	eICU	MIMIC-IV
0	4.79	5.35	4.67
1	4.29	2.27	4.06
2	11.53	10.98	10.40
3	6.33	4.65	6.76
4	6.02	4.48	7.83
5	5.10	6.10	6.14
6	13.62	22.09	11.26
7	8.15	13.73	6.88
8	7.45	5.87	7.23
9	7.37	8.67	6.93
10	0.06	0.16	0.08
11	1.75	0.65	1.51
12	3.73	0.68	4.36
13	0.56	0.02	0.59
14	7.29	6.16	5.72
15	4.68	5.08	6.64
16	7.30	3.05	8.95

Multi-label Classification Tasks Table VII Each row represents a class label, and the corresponding percentages denote the proportion of instances assigned to each class in the respective dataset. Class unification across the datasets follows DescEmb [25].

REFERENCES

[1] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun, "Doctor AI: Predicting clinical events via recurrent neural networks," in *Proc. Mach. Learn. Healthcare Conf.*, 2016, pp. 301–318.

[2] A. Awad, M. Bader-El-Den, J. McNicholas, and J. Briggs, "Early hospital mortality prediction of intensive care unit patients using an ensemble learning approach," *Int. J. Med. Inform.*, vol. 108, pp. 185–195, 2017.

[3] S. W. Thiel, J. M. Rosini, W. Shannon, J. A. Doherty, S. T. Micek, and M. H. Kollef, "Early prediction of septic shock in hospitalized patients," *J. Hosp. Med.: Official Pub. Soc. Hosp. Med.*, vol. 5, no. 1, pp. 19–25, 2010.

[4] K. Shameer et al., "Predictive modeling of hospital readmission rates using electronic medical record-wide machine learning: A case-study using Mount Sinai heart failure cohort," in *Proc. Pacific Symp. Biocomput.*2017, pp. 276–287.

[5] A. Ashfaq, A. Sant'Anna, M. Lingman, and S. Nowaczyk, "Readmission prediction using deep learning on electronic health records," *J. Biomed. Inform.*, vol. 97, 2019, Art. no. 103256.

[6] R. Miotto, L. Li, and B. Kidd, "Deep patient: An unsupervised representation to predict the future of patients from the electronic health records," *Sci Rep*, vol. 6, 2016, Art. no. 26094.

[7] D. J. Park, M. W. Park, H. Lee, Y.-J. Kim, Y. Kim, and Y. H. Park, "Development of machine learning model for diagnostic disease prediction based on laboratory tests," *Sci. Rep.*, vol. 11, no. 1, 2021, Art. no. 7567.

[8] I. Landi et al., "Deep representation learning of electronic health records to unlock patient stratification at scale," *NPJ Digit. Med.*, vol 3, no. 1, pp. 1–11, 2020.

[9] H. Harutyunyan, H. Khachatrian, D. C. Kale, G. Ver Steeg, and A. Galstyan, "Multitask learning and benchmarking with clinical time series data," *Sci. Data*, vol. 6, no. 1, 2019, Art. no. 96.

[10] H. Martínez Alonso and B. Plank, "When is multitask learning effective? Semantic sequence prediction under varying data conditions," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 44–53.

[11] T. J. Pollard, A. E. W. Johnson, J. D. Raffa, L. A. Celi, R. G. Mark, and O. Badawi, "The eICU collaborative research database, a freely available multi-center database for critical care research," *Sci. Data*, vol. 5, no. 1, pp. 1–13, 2018.

[12] A. Johnson, L. Bulgarelli, T. Pollard, L. A. Celi, R. Mark, and S. Horng, "MIMIC-IV, a freely accessible electronic health record dataset," *Sci. Data*, vol. 10, p. 1, 2023.

[13] A. E. W. Johnson et al., "MIMIC-III, A freely accessible critical care database," *Sci. Data*, vol. 3, no. 1, pp. 1–9, 2016.

[14] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzel, "Learning to diagnose with LSTM recurrent neural networks," presented at the Int. Conf. Learn. Representations, San Juan, Puerto Rico, May 2-4, 2016.

[15] P. Nguyen, T. Tran, N. Wickramasinghe, and S. Venkatesh, "Deepr: A convolutional net for medical records," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 1, pp. 22–30, Jan. 2017.

[16] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[17] H. Song, D. Rajan, J. J. Thiagarajan, and A. Spanias, "Attend and diagnose: Clinical time series analysis using attention models," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 4091–4098.

[18] J. Shang, T. Ma, C. Xiao, and J. Sun, "Pre-training of graph augmented transformers for medication recommendation," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, vol. 1, 2019, pp. 5953–5959.

[19] E. Choi et al., "Learning the graphical structure of electronic health records with graph convolutional transformer," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 606–613.

[20] S. Wang, M. B. A. McDermott, G. Chauhan, M. Ghassemi, M. C. Hughes, and T. Naumann, "MIMIC-extract: A data extraction, preprocessing, and representation pipeline for MIMIC-III," in *Proc. ACM Conf. Health, Inference, Learn.*, 2020, pp. 222–235.

[21] M. McDermott et al., "A comprehensive EHR timeseries pre-training benchmark," in *Proc. Conf. Health, Inference, Learn.*, 2021, pp. 257–278.

[22] A. Rajkomar et al., "Scalable and accurate deep learning with electronic health records," *NPJ Digit. Med.*, vol. 1, no. 1, pp. 1–10, 2018.

[23] J. C. Mandel, D. A. Kreda, K. D. Mandl, I. S. Kohane, and R. B. Ramoni, "SMART on FHIR: A standards-based, interoperable apps platform for electronic health records," *J. Amer. Med. Inform. Assoc.*, vol. 23, no. 5, pp. 899–908, 2016.

[24] Z. Wu, C. Xiao, L. M. Glass, D. M. Liebovitz, and J. Sun, "AutoMap: Automatic medical code mapping for clinical prediction model deployment," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2022, pp. 505–520.

[25] K. Hur, J. Lee, J. Oh, W. Price, Y. Kim, and E. Choi, "Unifying heterogeneous electronic health records systems via text-based code embedding," in *Proc. Conf. Health, Inference, Learn.*, 2022, pp. 183–203.

[26] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[27] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 12449–12460.

[28] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Ch. Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2018, vol. 1, pp. 4171–4186.

[29] P. Yin, G. Neubig, W.-T. Yih, and S. Riedel, "TaBERT: Pretraining for joint understanding of textual and tabular data," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, vol. 1, pp. 8413–8426.

[30] Z. Zhang, C. Yan, X. Zhang, S. L. Nyemba, and B. A. Malin, "Forecasting the future clinical events of a patient through contrastive learning," *J. Amer. Med. Inform. Assoc.*, vol. 29, no. 9, pp. 1584–1592, 2022.

[31] Y.-P. Chen, Y.-H. Lo, F. Lai, and C.-H. Huang, "Disease concept-embedding based on the self-supervised method for medical information extraction from electronic health records and disease retrieval: Algorithm development and validation study," "*J. Med. Internet Res.*, vol. 23, no. 1, 2021, Art. no. e25113.

[32] E. Steinberg, K. Jung, J. A. Fries, C. K. Corbin, S. R. Pfohl, and N. H. Shah, "Language models are an effective representation learning technique for electronic health record data," *J. Biomed. Inform.*, vol. 113, 2021, Art. no. 103637.

[33] Y. Li et al., "BEHRT: Transformer for electronic health records," *Sci. Rep.*, vol. 10, no. 1, pp. 1–12, 2020.

[34] L. Rasmy, Y. Xiang, Z. Xie, C. Tao, and D. Zhi, "Med-BERT: Pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction," *NPJ Digit. Med.*, vol. 4, no. 1, pp. 1–13, 2021.

[35] E. Choi et al., "Multi-layer representation learning for medical concepts," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1495–1504.

[36] L. Song, C. W. Cheong, K. Yin, W. K. Cheung, B. C. M. Fung, and J. Poon, "Medical concept embedding with multiple ontological representations," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 4613–4619.

[37] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, 2017.

[38] O. Press, N. A. Smith, and M. Lewis, "Train short, test long: Attention with linear biases enables input length extrapolation," presented at the Int. Conf. Learn. Representations, Virtual, Apr. 25-29, 2022.

[39] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "SpanBERT: Improving pre-training by representing and predicting spans," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 64–77, 2020.

[40] E. Alsentzer et al., "Publicly available clinical BERT embeddings," in *Proc. 2nd Clin. Natural Lang. Process. Workshop*, 2019, pp. 72–78, doi: 10.18653/v1/W19-1909.

[41] T. Gupta et al., "Sequential organ failure assessment component score prediction of in-hospital mortality from sepsis," *J. Intensive Care Med.*, vol. 35, no. 8, pp. 810–817, 2020.

[42] "HCUP clinical classifications software (CCS) for ICD-9-CM," Healthcare Cost and Utilization Project (HCUP), Agency for Healthcare Research and Quality, Rockville, MD, USA, 2016. [Online]. Available: https://hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp

[43] B. Gyawali, K. Ramakrishna, and A. S. Dhamoon, "Sepsis: The evolution in definition, pathophysiology, and management," *SAGE Open Med.*, vol. 7, 2019, Art. no. 2050312119835043.

[44] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou, "Word translation without parallel data," presented at the Int. Conf. Learn. Representations, Vancouver, Canada, Apr. 30-May 3, 2018.

[45] K. Choromanski et al., "Rethinking attention with performers," presented at the Int. Conf. Learn. Representations, Virtual, May 3-7, 2021.

[46] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," presented at the Int. Conf. Learn. Representations, Virtual, Apr. 25-29, 2022.