Medical Image Analysis 89 (2023) 102895

Contents lists available at ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

Unsupervised pre-training of graph transformers on patient population graphs

Chantal Pellegrini [a,*], Nassir Navab [a,b], Anees Kazi [a,c,1,2]

[a] Computer Aided Medical Procedures, Technical University of Munich, Munich, Germany
[b] Computer Aided Medical Procedures, Johns Hopkins University, Baltimore, USA
[c] Massachusetts General Hospital, Harvard Medical School, Cambridge, MA, USA

ARTICLE INFO

MSC:
41A05
41A10
65D05
65D17

Keywords:
Pre-training
Transformer
Population graphs
Outcome/disease prediction
Multi-modal data analysis

ABSTRACT

Pre-training has shown success in different areas of machine learning, such as Computer Vision, Natural Language Processing (NLP), and medical imaging. However, it has not been fully explored for clinical data analysis. An immense amount of clinical records are recorded, but still, data and labels can be scarce for data collected in small hospitals or dealing with rare diseases. In such scenarios, pre-training on a larger set of unlabeled clinical data could improve performance. In this paper, we propose novel unsupervised pre-training techniques designed for heterogeneous, multi-modal clinical data for patient outcome prediction inspired by masked language modeling (MLM), by leveraging graph deep learning over population graphs. To this end, we further propose a graph-transformer-based network, designed to handle heterogeneous clinical data. By combining masking-based pre-training with a transformer-based network, we translate the success of masking-based pre-training in other domains to heterogeneous clinical data. We show the benefit of our pre-training method in a self-supervised and a transfer learning setting, utilizing three medical datasets TADPOLE, MIMIC-III, and a Sepsis Prediction Dataset. We find that our proposed pre-training methods help in modeling the data at a patient and population level and improve performance in different fine-tuning tasks on all datasets.

1. Introduction

A large amount of medical data is collected on a daily basis in many different hospitals. Often this data is stored in Electronic Health Records (EHRs), digital patient charts containing various information such as the patient's medical history, diagnoses, treatments, medical images, and lab or test results. However, even though a large amount of data is recorded, for some tasks labeled data is scarce, as labeling can be tedious, time-consuming, and expensive (Xiao et al., 2018). Nevertheless, their digital and structured form can enable easy access to apply learning-based methods to EHR data, in particular unsupervised methods (Landi et al., 2020). Further, data collected in small hospitals or for rare diseases is often limited (Mitani and Haneuse, 2020). In these scenarios, the ability to leverage the large body of unlabeled clinical data could boost the performance and confidence of prediction systems on these smaller datasets. This, in turn, can support clinicians in better and faster diagnosis and decision-making.

Clinical records contain multiple heterogeneous data types, including imaging and non-imaging data. While non-imaging data usually is in numerical format, medical images consist of 2D or 3D data. In this work, we use imaging biomarkers to convert the imaging data to numerical features allowing a simple fusion of multi-modal input and reducing the computational requirements. This allows us a very general formulation of our method, which can be applied to both static and longitudinal data, handling both imaging and non-imaging features simultaneously. In our ablation studies, we further show that our model can be extended to deal with spatial images.

Unsupervised pre-training can be a useful tool to exploit unlabeled data and has shown great success in other domains, such as natural language processing (NLP) (Mikolov et al., 2013; Pennington et al., 2014; Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019; Yang et al., 2019; Liu et al., 2019), computer vision (CV) (Pathak et al., 2016; Bojanowski and Joulin, 2017; Caron et al., 2018; Komodakis and Gidaris, 2018; Bao et al., 2021) and medical imaging (Bai et al., 2019;

* Corresponding author.
E-mail address: chantal.pellegrini@tum.de (C. Pellegrini).

[1] The work was done while A. Kazi was affiliated with the TU Munich.

[2] One of the datasets used in preparation of this article was obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf.

https://doi.org/10.1016/j.media.2023.102895
Received 25 July 2022; Received in revised form 6 July 2023; Accepted 7 July 2023
Available online 11 July 2023

Chen et al., 2019; Ouyang et al., 2020). In these domains various types of pre-training have been proposed, including generative approaches e.g. based on auto-encoders, contrastive learning, and the application of hand-crafted pre-text tasks such as masked content prediction (Liu et al., 2021; Han et al., 2021). In the medical domain, next to medical imaging, pre-training was applied for instance on medical code data (Shang et al., 2019; Li et al., 2020; Rasmy et al., 2021; Pang et al., 2021) and textual EHR data (Park et al., 2022). However, it is not explored enough for complex clinical data such as heterogeneous, multi-modal patient records (EHRs). Only a few previous works investigated pre-training for this type of data, laying the ground-stone for pre-training on heterogeneous EHRs (McDermott et al., 2021; Gupta et al., 2020). They show the benefit of pre-training, especially for scenarios with limited labeled data. Nevertheless, their improvements via pre-training are limited, in particular, if more labeled data is available, and their pre-training task designs do not fully take the longitudinal and complementary nature of clinical records into account.

In the papers mentioned above the community explored various model architectures, including RNNs, GRUs, and Transformers. For sequential data, such as natural language and time-series data, transformer models (Vaswani et al., 2017) are currently dominant. These models often are combined with self-supervised masking-based pre-training tasks, which has proven to be a promising combination (Devlin et al., 2019; Bao et al., 2021; Li et al., 2020). In this work, we focus on this type of self-supervised pre-training and combine it with a graph-transformer-based network.

In the recent literature, a new way for multi-modal patient data analysis using patient population graphs is getting explored. In a population graph, each node n_i in the graph G represents a patient and the edges in G incorporate the similarities between the patients. Such patient population graphs have been leveraged to help analyze patients' medical data using the relationship among different patients (Ahmedt-Aristizabal et al., 2021). This enables clinically semantic modeling of the data. By using the patient population graph during unsupervised pre-training, node representations based on feature similarities between the subjects can be learned. These representations can improve the understanding of the data, which can then help to improve patient-level predictions. Therefore we model the EHR data in a patient population graph for both pre-training and fine-tuning.

Several works successfully apply pre-training to graph data and show the benefits on both common graph benchmarks (Zhang et al., 2020; Hu et al., 2020b) as well as domain-specific data such as molecular or biological graphs (Hu et al., 2020a; Rong et al., 2020; Lu et al., 2021). However, these methods are often specialized to a certain domain, such as protein or molecule graphs, or focus on improving the graph-level embedding. To the best of our knowledge, pre-training was not previously applied to patient population graphs, where the pre-training task needs to be formulated to foster meaningful node-level embeddings, representing the patient's data. As patient population graphs have proven useful for outcome and disease prediction tasks on clinical data, we believe a well-working pre-training technique for patient population graphs is of high value to the community and has the potential to improve performance in many patient-level prediction tasks.

In this paper, we propose a graph transformer-based model suitable for learning on multi-modal clinical data in form of population graphs. Motivated by the huge success of masked language modeling pre-training for transformer models, we propose masking-based pre-training methods and combine them with our graph transformer-based architecture. Our pre-training methods are specialized for multimodal clinical data. Our main contributions are:

- We develop multiple novel unsupervised pre-training methods based on masked imputation of clinical input features, which are specifically designed for (longitudinal) EHRs and multi-modal clinical data modeled as population graphs.
- We propose a novel (graph) transformer-based model suitable to learn over heterogeneous clinical data allowing multi-modal data fusion. The intelligent design and combination of state-of-the-art building blocks allow us to deal with heterogeneous data and show the potential of transformers for multi-modal clinical data analysis. Our model is designed to handle various input data types occurring in multi-modal clinical records, taking static and time-series data and continuous as well as discrete numerical features into account.
- By combining our graph transformer-based model with masking-based pre-training, we show a way to translate the success of pre-training of transformers in other domains to multi-modal clinical data.
- We show significant performance gains through pre-training when fine-tuning with as little as 1% and up to 100% labels in both the self-supervised and the transfer learning setup, providing a solution to limited labeled data.

We evaluate our method over general EHR data as well as brain imaging data over two publicly available, medical datasets, MIMIC-III (Johnson et al., 2016) and TADPOLE (Marinescu et al., 2018) for Length-of-Stay prediction (Zebin et al., 2019; Wang et al., 2020; McDermott et al., 2021), combined Discharge and Mortality Prediction (McDermott et al., 2021) and Alzheimer's disease prediction (Parisot et al., 2018; Kazi et al., 2019b; Cosmo et al., 2020). We provide an extensive analysis of our model and the effects of the different pre-training approaches on fine-tuning performance for different amounts of labels. Further, we test our method in a transfer learning setting, where we pre-train on the MIMIC-III dataset and fine-tune on the Sepsis Prediction dataset published in the PhysioNet/Computing in Cardiology Challenge 2019 (Reyna et al., 2019; Goldberger et al., 2000). Our source code is available at https://github.com/ChantalMP/Unsupervised_pre-training_of_graph_transformers_on_patient_population_graphs.

The remainder of this paper is structured as follows. After presenting the most relevant related work in Section 2, we provide a detailed description of our method in Section 3, including population graph construction, the proposed model architecture, and the proposed pre-training strategies. In Section 4 we first introduce the datasets we use and our experimental setup and then show our results, as well as their interpretation and discussion. Finally, a conclusion is given in Section 5.

2. Related work

The main contribution of our method lies in investigating unsupervised pre-training for heterogeneous EHRs modeled as population graphs for patient-outcome prediction. Accordingly, we divided this section into multiple parts focusing on patient population graphs and pre-training for graph and EHR data.

2.1. Population graphs for patient outcome prediction

Several previous works use patient population graphs in combination with graph neural networks in the field of diagnosis and patient outcome prediction (Ahmedt-Aristizabal et al., 2021). Parisot et al. (2018) introduced the use of GCNs for the analysis of population graphs in the medical domain for combining imaging and non-imaging data. Given a population of patients, they construct a fully connected graph, where the edges are based on the similarity of non-imaging data, while the node features describe a patient's imaging data. InceptionGCN (Kazi et al., 2019a) introduces an inception module for spectral GCNs, showing one way to deal with heterogeneous graphs to model multi-modal data. Valenchon and Coates (2019) propose to use multiple graphs based on different subject features. In another work, Kazi et al. (2019b) propose to use an attention mechanism for weighting the subject's demographic features. Several works aim to

improve the graph structure by either updating a pre-constructed graph during training (Huang and Chung, 2020) or learning the optimal graph end-to-end (Cosmo et al., 2020; Kazi et al., 2022). Other works focus on dealing with missing (Vivar et al., 2018) or imbalanced data (Ghorbani et al., 2022). Overall, modeling clinical data in a population graph has been proven a promising direction in patient outcome prediction.

2.2. Masking-based pre-training

In the NLP domain, pre-training was adopted mainly using self-supervised learning, aimed to understand the intrinsics of natural language without the need for any human supervision (Qiu et al., 2020). This allows to exploit the huge corpora of unlabeled text data. BERT (Devlin et al., 2019) is one of the most important works about pre-training transformer-based language models. It is designed to learn bidirectional representations from unlabeled text. After being fine-tuned with task-specific data, it has set a new state-of-the-art in many NLP tasks. They introduce masked language modeling (MLM), which until now is one of the most used pre-training tasks for NLP and was further adapted to be used in several other domains (Yu et al., 2021; Bao et al., 2021; Li et al., 2020). Given an input text, during MLM, 15% of the input tokens are randomly masked and replaced with a mask token '[MASK]'. After being processed by several transformer layers, the final hidden representations are fed into an output softmax over the defined vocabulary aiming to predict the masked token. For fine-tuning, the model is initialized with the pre-trained weights, only the final prediction layers are changed according to the task. In the medical domain, BERT-like pre-training was applied for tasks such as disease prediction (Rasmy et al., 2021), medication recommendation (Shang et al., 2019), medical imaging (Wang et al., 2021), clinical outcome prediction (Pang et al., 2021; McDermott et al., 2021) and clinical NLP tasks (Alsentzer et al., 2019; He et al., 2020). However, it was only rudimentary explored for heterogeneous EHR data.

2.3. Pre-training on graph data

The previously proposed pre-training techniques for graph data reach from node-level to graph-level to generative tasks. These tasks include tasks like attribute reconstruction, graph-level property prediction, or auto-regressive generation of nodes and edges (Xie et al., 2022). Hu et al. (2020a) propose to combine node and graph level pre-training by first using a node-level task such as attribute masking to train the model, followed by training to predict graph level properties (e.g. properties of a certain protein). Rong et al. (2020) propose a transformer-based network, pre-trained by predicting properties of masked sub-graphs (such as the number of neighbors of a certain atom type) and the presence of certain motifs (recurrent sub-graphs such as functional groups in molecules) in a graph. Zhang et al. (2020) propose Graph-BERT, a translation of BERT pre-training to the graph domain. Instead of processing a full graph, they create link-less sub-graphs of a node's neighborhood, add certain positional embeddings to each node, and process them by a conventional transformer. As pre-training tasks, they propose to reconstruct a node's attributes using a linear layer and to optimize the model to have high cosine-similarities between the embeddings of nodes with a high ground truth intimacy. Hu et al. (2020b) propose a generative graph pre-training framework called GPT-GNN. For pre-training, both attributes and edges are generated, given a partial graph as initialization. While the previous works focus on unsupervised pre-training within one graph domain, Verma and Zhang (2019) investigated transfer learning for graph data. Their model consists of an input transformer, a simple linear layer, a graph encoder, which is a conventional GNN, and a task-specific graph decoder. While the input transformer and the graph decoder are dataset-specific, the graph encoder is trained jointly on several graph datasets from the bioinformatics and social network domain. They improve classification results in comparison to classical GNNs and graph kernel methods. The aforementioned methods could show the applicability of pre-training in the graph domain. However, to the best of our knowledge, no previous work investigated pre-training for patient population graphs.

2.4. Pre-training on EHR data

The most general form of multi-modal clinical records are Electronic Health Records, which can contain any data recorded over a patient's stay in a hospital. As EHRs are usually recorded throughout the patient's stay, they often have a sequential structure, making sequence models like transformers a good fit to learn on this data. Some previous works study how to pre-train BERT (Devlin et al., 2019) over simplified EHR records for downstream tasks in disease and medication code prediction. Here the input records contain only sequences of medical codes, such as diagnosis or medication codes but do not include multi-modal heterogeneous data. In many of these works, an adapted version of MLM is used as a pre-training task (Shang et al., 2019; Li et al., 2020; Rasmy et al., 2021; Pang et al., 2021). Agrawal et al. (2022) investigate order-contrastive pre-training for clinical time-series data and test their method on synthetic data and temporally ordered clinical radiology notes. McDermott et al. (2021) propose one pre-training approach over heterogeneous EHR data. They create a pre-training benchmark over the eICU (Pollard et al., 2018) and MIMIC-III (Johnson et al., 2016) datasets and come up with two baseline pre-training methods, unsupervised masked imputation, where random time points are masked, and supervised multi-task learning. Furthermore, they define several downstream tasks to evaluate their method. Pre-training and fine-tuning are performed over records of single patient stays in the ICU using a bi-directional Gated Recurrent Unit. Gupta et al. (2020) present an approach for transfer learning to improve outcome prediction on longitudinal EHR records by processing each input feature separately with a pre-trained TimeNet (Malhotra et al., 2017), the encoder of an auto-encoder RNN, which was pre-trained to reconstruct diverse univariate time-series data. The aforementioned methods have one common drawback, as they often simplify the given EHR data in order to apply pre-training. The few methods working with heterogeneous EHRs, do still not fully consider the form of EHR data in their pre-training task design and show only limited improvements.

For the graph domain, several works show the promise of pre-training. However, no work investigated pre-training for patient population graphs, where it can lead to better patient outcome predictions. On the other hand, work on pre-training for EHR data remains very limited. The existing previous methods do not fully account for their longitudinal and heterogeneous nature and show limited improvements. We improve upon these works and propose novel EHR-specific pre-training techniques based on masking for EHR data modeled as population graphs. In the next section, we explain our method in detail.

3. Method

Our method is designed to target patient-level prediction tasks on a dataset $\mathbf{D}$, composed of the clinical records of N patients. Toward this, we propose a two-step pipeline. The first step entails the use of unsupervised pre-training methods to enhance the general understanding of the data by our model. In the second step, we fine-tune the pre-trained model on a downstream prediction task T. The understanding learned via unsupervised pre-training aims to improve the performance on downstream tasks despite limited labeled data.

In the dataset $\mathbf{D}$, let n_i and r_i be the ith patient and its corresponding record, where $i \in [1, N]$. Every record $\mathbf{r_i}$ consists of a set of heterogeneous input features, denoted by $\mathbf{r_i} \subseteq [\mathbf{d_i}, \mathbf{c_i}, \mathbf{t_{d_i}}, \mathbf{t_{c_i}}]$. Each record can contain all or a subset of these features. The feature types include static discrete features $\mathbf{d_i} \in \mathbb{N}^D$, static continuous features $\mathbf{c_i} \in \mathbb{R}^C$, and discrete as well as continuous time-series features $\mathbf{t_{d_i}} \in \mathbb{R}^{S_d \times \tau}$ and $\mathbf{t_{c_i}} \in \mathbb{R}^{S_c \times \tau}$, where $S_{d/c}$ denotes the feature dimension and τ the length of the time-series. Static features $\mathbf{d_i}$ and $\mathbf{c_i}$ encompass different numerical values such as patient demographics or measurements taken once for every patient, whereas the time-series features $\mathbf{t_{d_i}}$ and $\mathbf{t_{c_i}}$ consist of a sequence of values recorded throughout a patient's stay in the hospital, such as vital signs (e.g. heart rate, temperature) or

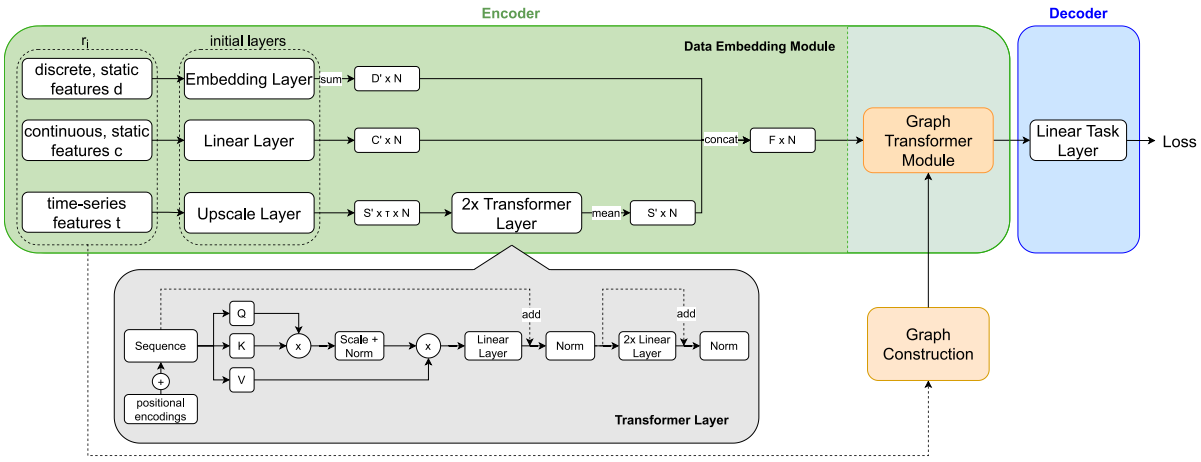

Fig. 1. Overview of the proposed architecture. All input features are combined into one node embedding, applying transformer layers to enhance the time-series features. The resulting graph is processed by several Graphormer layers and a linear task layer. The population graph is constructed using the original input features r_i.

repeatedly measured lab values (e.g. Hemoglobin, glucose level). All targeted downstream tasks T are binary or multi-class classification tasks on a patient level with labels $\mathbf{Y} \in \mathbb{N}^L$ given for L classes, and the task is to make predictions for all patient records in the test set.

As the same model is used for pre-training and fine-tuning, to understand our pipeline we will start by introducing the proposed model architecture including the different sub-modules and the population graph construction. Afterward, we give details about the pre-training and fine-tuning procedure.

3.1. Model architecture

Our model is tasked with making node-level predictions for every node in the graph G, based on the input features $\mathbf{d}, \mathbf{c}, \mathbf{t_d}$ and $\mathbf{t_c}$. For this task, we propose a model, consisting of an encoder and a decoder. The encoder is used to generate meaningful node representations given the input EHR graph. Thereafter the decoder's task is to capture the essence of features inclined toward a node-level classification task. The same model is used in the pre-training as well as the fine-tuning stage.

3.1.1. Encoder

The encoder comprises a data embedding module and a graph transformer module, based upon Graphormer (Ying et al., 2021). The data embedding module is responsible to fuse the heterogeneous input features occurring in multi-modal clinical records. The fused features are then used as node embeddings in our constructed population graph which is processed by the graph transformer module.

Data embedding module:

The data embedding module converts the heterogeneous input features, $\mathbf{d}, \mathbf{c}, \mathbf{t_d}$ and $\mathbf{t_c}$, into one common representation to form node embeddings for every node in the graph.

To process static, discrete input features, we follow the conventional Graphormer (Ying et al., 2021) and apply an embedding layer, followed by a summation over the feature dimension. This transforms the input features $\mathbf{d_i}$ into embedded features $\mathbf{d'_i} \in \mathbb{R}^{D'}$.

While Graphormer is limited to static, discrete input features only, we extend the model to support static, continuous input features as well. As for continuous features embedding layers are not applicable, these features are processed by a linear layer. Again this results in an embedding vector $\mathbf{c'_i} \in \mathbb{R}^{C'}$.

Next to static features, we also support discrete and continuous time-series features $\mathbf{t_{d/c_i}} \in \mathbb{R}^{S_{d/c} \times \tau}$. We first upscale the feature dimension of every time step by applying an upscale layer U_θ with weights θ. For discrete features, this upscale layer is an embedding layer and for continuous features, a linear layer, followed by a summation over the feature dimension. These up-scaled features $\mathbf{t^{(u)}_{d/c_i}}$ are further processed by two transformer layers, as described by Vaswani et al. (2017), to enhance the embeddings using temporal context. The transformer layers output adapted embeddings $(e_1, e_2, \dots, e_\tau)$ per time-step. The final embedding for the time-series features $\mathbf{t'_{d/c_i}}$, is then formed as the mean of the time-step embeddings $(e_1, e_2, \dots, e_\tau)$, which allows working with sequences of variable lengths. This results in embedded features $\mathbf{t'_{d_i}} \in \mathbb{R}^{S'_d}$ for the discrete and $\mathbf{t'_{c_i}} \in \mathbb{R}^{S'_c}$ for continuous features.

The feature vectors $\mathbf{d'_i}, \mathbf{c'_i}$, $\mathbf{t'_{d_i}}$ and $\mathbf{t'_{c_i}}$ can now be concatenated to form the final node embeddings $n_i \in \mathbb{R}^F$ for each of the N nodes, where $F = \sum_{F_k \subseteq [D', C', S'_d, S'_c]} F_k$ is the sum of the feature dimensions of the transformed input features.

Graphormer Module: The backbone of our model comprises **L** graph transformer layers as proposed by Ying et al. (2021). Dependent on the dataset we vary the number of Graphormer layers. Graphormer is based upon transformer (Vaswani et al., 2017) and outperforms traditional graph neural networks in several graph-level prediction tasks. Their main contribution is the encoding of the graph structure, without restricting attention to neighborhoods. Instead, they attend to all nodes in the graph and incorporate the graph structure via structural encodings, which encode the in and out degrees of the nodes, the length of the shortest path between each node pair, and the edge features lying on this path. The encoding of the in and out degree is added directly to the node features of every node, while the other structural encodings are added as a bias to the attention score between two nodes. Fig. 2 shows an overview of a Graphormer layer.

The input graph for Graphormer is constructed based on the features of the input records r_i as described in the following paragraph.

Graph Construction: Let $G = (V, E)$ be a population graph where the nodes $V = \{n_1, \dots, n_N\}$ represent the patients. Each node is associated with features coming from the patient record r_i, and the edges $E = \{e_{i,j} : i, j \in N\}$ define the connections between the nodes. For each node pair n_i and n_j representing the records r_i and r_j, we calculate a similarity score $S(r_i, r_j)$ between the features to decide if there exists an edge $e_{i,j}$. As we have various feature types, we define different similarity scores, which are suitable for the respective features, based on L2 distance for continuous features and absolute matching for discrete features (Ahmedt-Aristizabal et al., 2021). After computing the similarities for all feature types we average them to get one overall similarity score for each record pair. Before averaging, all type-specific similarity scores are normalized to a range between zero and one using min–max normalization or sigmoid. As our focus does not lie on graph construction, we choose a conventional graph construction method and construct a k-NN graph with $k = 5$. The value of k is

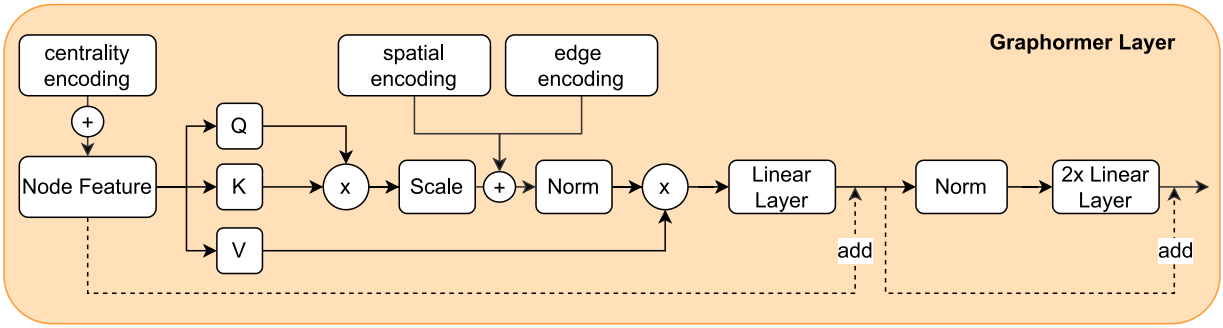

Fig. 2. Overview of a Graphormer layer as proposed by Ying et al. (2021). Graphormer is a graph transformer model based on global attention to all nodes. The graph structure is integrated via different structural encodings. The centrality encoding, which encodes the in and out degree of each node is added directly to the node features. Given the shortest path between two nodes, the spatial and edge encoding encode the length and the edge features of the path. They are both added as a bias to the attention.

chosen experimentally to avoid many disconnected components as well as very densely connected regions (see supplementary material). We want to avoid disconnected as well as very densely connected regions, as in both extremes the graph structure does not give any valuable information about the relationships between the nodes. This similarity score is additionally used as the edge feature of the edge between two nodes. This enables the model to give more attention to nodes connected over a path with high feature similarities. The experiment section contains a detailed description of the graph construction per dataset.

3.1.2. Decoder

The decoder consists of one or multiple linear layers for predicting outputs for the current pre-training or downstream task. Depending on the task, the decoder layer is adapted to fit the desired output dimensions. When training for an end-task the decoder is a linear output layer, with an output dimension according to the number of classes in the targeted task. During pre-training, the decoder also consists of linear layers, one for each input data type. This layer predicts an output vector in the same shape as the input data, where each element can be interpreted as the prediction for the corresponding input value. Even though we make a prediction for all input values, only the predictions of masked values are used to update the model.

3.2. Training pipeline

Until now we described the model architecture needed for fine-tuning and pre-training and its components. Now we will explain the two main steps of our pipeline, pre-training and fine-tuning, in detail.

3.2.1. Pre-training step

We propose several unsupervised pre-training strategies for patient population graphs of Electronic Health Records. All proposed pre-training methods use the paradigm introduced with masked language modeling and task the model to predict masked features or attributes derived from them. We design our masking techniques toward capturing different aspects of clinical record data. As clinical records often contain longitudinal data, several of our proposed methods are focused on this scenario, however, we also evaluate a version suitable for static data. For all methods, masking is performed by replacing certain feature values with a fixed value. Therefore, let $F \in \mathbb{R}^{\square x \tau}$ denote a feature matrix and $M \in \mathbb{R}^{\square x \tau}$ denote the corresponding mask, where $\square$ corresponds to the feature dimension of the current feature matrix and τ to the length of the time-series. For static features $\tau = 1$. The dimensions depend on the considered input features, which can be any of $\mathbf{d}, \mathbf{c}, \mathbf{t_d}$ and $\mathbf{t_c}$. M is defined as follows, where $f_{i,j}$ is the feature value at position (i, j):

$$M_{i,j} = \begin{Bmatrix} 0 \; f_{i,j} \text{ is masked} \\ 1 \text{ else} \end{Bmatrix} \tag{1}$$

For all static features, the masked features F' are then computed as an element-wise product of the mask with the feature matrix (Eq. (2)). This leads to masked values being set to zero.

$$F' = F \circ M \tag{2}$$

For discrete features, a 'mask token' $m \in \mathbb{N}$ is used to replace masked elements (Eq. (3)). The mask token m has an arbitrarily chosen, but fixed value. The mask token does not influence the performance, as the discrete masked input is processed by an embedding layer, which functions as a map from feature values to embedding representations.

$$F' = F \circ M + m * (1 - M) \tag{3}$$

For time-series features, we further add a binary column per feature to the input vector, that encodes which hours in the time-series are masked. We optimize the model using (binary) cross-entropy loss (ml glossary, 2022a) for the prediction of discrete features and mean squared error loss (ml glossary, 2022b) for predicting continuous features.

To mask the feature values we propose different masking strategies for both static as well as time-series input data:

Feature Masking Given records with an arbitrary number of features for all patients, we randomly mask a fixed percentage of the features, called 'masking ratio', for every record $\mathbf{r_i}$ in the training set. The model is optimized to predict these masked values. The masking ratio is chosen experimentally. This pre-training technique aims to directly teach the model how the patient's features relate to one another, by optimizing it to predict some masked features while having access to the remaining features of the patient. To this end, the masking ratio should be smaller than 100%, such that only a subset of the features of each patient is masked.

Static Feature Masking (SFM) For static patient data, every patient feature encompasses only one value. Several of these features are randomly selected to be masked and predicted.

Time-Series Feature Masking (TFM) When dealing with time-series features, a completely random selection of feature values to mask would distribute the masked values over the features as well as the time steps. As time-series features in clinical data often encompass repeated or only slowly changing measurements like vital signs or treatment features, feature values of neighboring time points tend to be similar. Knowing the previous and future value of a feature can make predicting random features too easy. An overly simple task has little value for pre-training, as the model does not have to develop a real understanding of the data to solve it. Thus for Time-Series Feature Masking, we mask all time points of an experimentally chosen percentage of each node's features. In this way, the model can neither see previous nor future values to infer the masked feature, but rather has to rely on other features and data from other patients.

Block-wise Masking (BM) Instead of masking all time points of the given feature time-series, we randomly mask a block of H hours. Again

the masking ratio, which determines how many features are masked, is experimentally chosen. Here, the model can access previous and future values of the same feature to make a prediction, but we choose a block size $H > 1$ to avoid knowing all neighboring feature values. The goal of this pre-training method is to teach the model an understanding of the temporal context within a patient's stay.

Treatment Prediction (TP) Which treatments a patient receives is directly correlated to his condition and its progression. Doctors use the measurements taken from a patient, such as his vital signs and results of disease-specific tests, to decide on the next action in the treatment of the patient. In the health record of a patient the treatments which he receives, e.g. medications that are given, are often recorded. With the Treatment Prediction pre-training, the model is optimized to predict the treatments a patient received, which could help to learn to understand the disease of a patient, given the results of performed measurements. In this task, the model needs to predict a binary label per treatment, indicating if a patient received a certain treatment or not. In case the treatments are given as time-series, we reduce them to binary indicators to generate the labels. In the input data, we mask all treatment features.

Patient Masking (PM) All previously described pre-training tasks focus on predicting a subset of features of one patient or attributes derived from them. To solve this task it can be sufficient to concentrate on the unmasked feature values of the same patient. With our last pre-training method, we aim to encourage the model to take the other nodes in the graph into account. Toward this goal, we mask a random subset of patients in the graph. For these randomly selected patients, we mask all measurement and treatment features and then again optimize the model to predict these masked features. The only information that remains unmasked is the patient's demographics. Again the percentage of masked patients is decided via experiments.

Unsupervised Multi-Task Pre-training All of the proposed pre-training strategies have different prediction targets and aim to teach the model distinct knowledge about the graph and the patient features. Thus, combining them during pre-training can lead to a further enhanced and comprehensive understanding of the data. Moreover, training on all tasks simultaneously might have a regularizing effect on the training and can lead to less overfitting. To combine the pre-training tasks, for training we sample one task per batch, mask the samples in the batch correspondingly, make a prediction and then update the model with the corresponding loss. For the next batch, another pre-training task will be sampled. For this combined pre-training task, we need to adapt our decoder to be able to deal with multiple tasks at once. We add a decoder layer for every pre-training task to the model. Depending on which task was sampled the corresponding layer is used and updated. To select the best model we collect all validation predictions of one epoch and average the performance metrics of the different tasks.

All pre-training tasks are sampled with equal probability. Our experiments showed, that even with this straightforward task combination multi-task pre-training is superior on all datasets and tasks. Optimizing the sampling ratio per dataset could potentially improve the effect further, however it would increase the effort needed to apply our method to a new dataset.

3.2.2. Fine-tuning step

After pre-training the model with one of the tasks proposed in the previous section, the pre-trained model is then fine-tuned for a downstream task T.

For **Self-supervised Learning** the encoder of the model for fine-tuning is initialized using the weights learned during pre-training, while the decoder is initialized randomly.

For **Transfer Learning** the initial encoder layers of every feature type (see Fig. 1) cannot be initialized with the pre-trained weights, as the feature dimensions do not match. For example, if the pre-training dataset has N and the fine-tuning dataset M continuous static input features, the linear layer extracting these features needs to have an input dimension of N or M respectively. Therefore these layers are also initialized randomly, together with the decoder. The rest of the encoder, including the transformer and graph transformer layers, are pre-trained as for self-supervised learning.

In the following, we present our experiments, where we analyze the performance of our model and the effect of the proposed pre-training methods. To this end, we consider the performance improvement in different downstream tasks T after pre-training. We show results for both the self-supervised as well as the transfer learning setting.

3.2.3. Handling of missing data

When working with EHR data, missing values are very common. They can be caused by lack of collection or documentation, false reporting, or irrelevance of certain values for a given patient (Wells et al., 2013). For time-varying features, which are recorded multiple times over a patient's stay, another reason for the missingness is, that these features are not measured regularly each hour, but at a different amount of times, reaching from every two hours to once a day. In a data pre-processing step, we impute these missing values, to create a homogeneous input for our network, where every patient has the same amount of features and measurements per feature. In detail, we carry the first and last known value of a time-series forward and backward respectively, while all values in between are linearly interpolated. For features without any value for a patient, we use the mean of this feature from the training set for all time steps. With this technique, the interpolated values provide a meaningful estimation of the missing values. During the pre-training stage, we compute the reconstruction loss solely based on measured values, to avoid optimizing the model to predict possibly incorrect interpolated values. Overall, this enables our method to deal with missing values during the pre-training as well as the fine-tuning stage.

4. Experiments

To evaluate our method we use three medical data sets and evaluate with five different downstream tasks. In the following section, we first describe the used datasets and then present and discuss our results and ablation studies.

4.1. Datasets description

We use the two public data sets MIMIC-III (Johnson et al., 2016) and TADPOLE (Marinescu et al., 2018), both medical datasets containing multi-modal patient data, for evaluating our method in a self-supervised setting. Additionally, we use the Sepsis Prediction dataset from the PhysioNet/Computing in Cardiology Challenge 2019 (Reyna et al., 2019; Goldberger et al., 2000) to test our model in a transfer learning setup. The datasets are of different sizes and contain different types of input features, allowing for a comprehensive evaluation of our method.

4.1.1. MIMIC-III

As our first dataset we use MIMIC-III (Johnson et al., 2016), which is a large dataset, including the EHRs for ICU stays of over 50,000 patients. We use the pre-processed dataset provided by McDermott et al. (2021), which contains the data of ICU stays of 21.88K patients. This data only includes adult patients (at least 15 years old), who stayed one full day (24 h) or longer in the ICU. Each record contains three types of features:

- Demographics: age (static, continuous), gender, admission type, first care unit (static, discrete)
- Measurements: 56 different measurements, containing recordings of bedside monitoring and lab test results in hourly granularity (time-series, continuous). Missing values are frequent, as not every measurement is taken every hour. We impute these missing values with linear interpolation in a pre-processing step.

Table 1
MIMIC-III dataset statistics.

No. of samples	No. of features	No. of features for graph
21.88K	76	56

Table 2
MIMIC-III task statistics.

Task	No. of classes	Occ. Majority class
LOS	2	51.81%
ACU	18	25.24%
ACU-4	4	38.37%

- Treatments: binary features per hour, denoting for 16 different treatments, if they were applied in each hour (time-series, static)

For every patient, we only use data from the first 24 h of the stay, as our targeted downstream tasks are defined on this input window. To make our results comparable to previous work, we keep the downstream task definitions and the restriction to the input window of 24 h. All continuous features in this dataset are normalized to the normal distribution $N(0,1)$. We target three different downstream classification tasks: Length-of-Stay (LOS) and Final Acuity prediction (ACU) as defined in McDermott et al. (2021), as well as an adapted acuity prediction task defined in this work. We propose this adapted task because the original acuity prediction task contains a lot of very hard-to-distinguish classes (e.g. Federal Hospital vs. Short Term Care Hospital) as well as classes occurring very rarely (down to one sample in the whole dataset). Our adapted task (ACU-4) summarizes the 18 classes of the original task into four meaningful groups. The three downstream tasks are defined as follows:

Length-of-Stay Prediction (LOS) For this task, the goal is to predict if a patient will stay longer than three days or not given the data from the first 24 h of his stay. This can be considered a 2-class classification problem.

Final Acuity Prediction (ACU) Here the task is to predict a combination of multiple outcomes including the patient's mortality, the discharge location, and the place of death. All together form a multi-class prediction task with 18 classes given below:

Discharge Locations: Long Term Care Hospital, Rehab/Distinct Part of Hospital, Transfer to Cancer or Children Hospital, Home Health Care, Short Term Hospital, Intermediate Care Facility (Icf), Transfer to Federal Hospital, Transfer to PsychHospital, Home, Left Against Medical Advice, Home With Home Iv Provider, Other Facility, Hospice-Medical Facility, Skilled Nursing Facility (Snf), Hospice-Home, Snf-Medicaid Only Certified

Death Locations: In-ICU, In-Hospital

Final Acuity Prediction Adapted (ACU-4) Again the task is to predict whether the patient will be discharged, and if yes to where, or die. However, we reduced the problem to the following four classes: discharge to home, discharge to home but with additional health care, discharge to care facility, and death.

The total 76 features of MIMIC-III are used as input features, and a subset of them is further used for graph construction. Table 1 provides the main statistics about the MIMIC-III dataset and Table 2 about the different tasks.

Graph Construction: As MIMIC-III contains a large number of patients, it is computationally infeasible to construct a graph containing all patients and process it with our model. Thus, we randomly split the samples into groups and form sub-graphs with 500 patients each, which fit into memory. Table 3 shows the performance of our model in the LOS task for different graph sizes. While the performance with 250 and 500 nodes is very similar, using a graph size of 750 nodes leads to a performance drop. This indicates that the restriction to a limited graph size does not harm performance, rather a too-large graph can even have a negative impact. We hypothesize that a too-large graph size can make it harder to learn a meaningful attention distribution as the attention will be distributed over more nodes. At the same time, the gain of relevant new information is limited, as already a large number of similar patients are included in the graph. Every sub-graph contains train, validation, and test patients. By splitting randomly we avoid making assumptions about which patients the model should see at once. Instead, we provide a diverse set of patients in all graphs and allow our model to learn how to attend to these patients via the attention mechanism in the graph transformer. For every subset, we compute the similarity between each pair of patients. As MIMIC-III contains a large number of heterogeneous features, we chose the features for graph construction experimentally. Table 4 shows the performance in LOS prediction on the first fold of MIMIC-III when using either all or only one specific type of feature for graph construction. Using a graph based on the measurement features proved to be superior, thus we use a similarity computed over the measurement features to construct our population graph. To be able to handle a different amount of measurements in each time-series, we do not directly compare the hourly features, but instead construct feature descriptors $f_d = (mean(f), std(f), min(f), max(f))$ per patient and feature for each of the 56 measurement features. We then compute the average similarity over all feature descriptors f_d between two patients r_i and r_j:

$$Sim(r_i, r_j) = \frac{\sum_{f \in f_d} \| f_{r_i} - f_{r_j} \|}{|f_d|} \tag{4}$$

Given the similarities between all patient pairs for a sub-graph, we construct a k-NN graph with k = 5. This leads to on average 18 ± 5.1 disconnected components in the graph.

Pre-Training Configuration: On MIMIC-III, we mask measurement and treatment data from the first 24 h of each patient's stay. As described before, we compute the loss only over measured features, which are inherently included in the data, and exclude the missing values, which were interpolated in the pre-processing step. We compare all of our proposed pre-training methods, which are designed to handle time-series and treatment data. The masking configurations are given as follows:

Time-Series Feature Masking (TFM): masking of 30% of the features;

Block-wise Masking (BM): masking of six-hour blocks in 100% of the features;

Treatment Prediction (TP): all treatments are masked;

Patient Masking (PM): masking of 10% of the patients;

Multi-Task (MT): We combine all four tasks (TFM, BM, TP, and PM). For all tasks, we use the same configuration as when trained alone. For all pre-training tasks, we determined the masking ratios experimentally pre-training with several masking ratios and fine-tuning on LOS prediction. For the further tasks on MIMIC-III, we keep the same masking ratios to be able to use the same pre-trained model. The results of this experiment are shown in Table 5.

Table 3
Validation performance of our model with different sub-graph sizes for MIMIC-III. For computational reasons, we performed this test only on one fold.

	250 nodes	500 nodes	750 nodes
AUC	77.83	77.38	73.19
ACC	71.10	70.96	67.85

Table 4
Performance of our model with different graph constructions for MIMIC-III. For computational reasons we performed this test only on one fold.

	Age	Treatments	All	Demographics	Measurements
AUC	74.79	75.20	75.20	75.21	**77.40**
ACC	69.40	69.12	69.59	69.82	**71.19**

Table 5
Experiments to find the optimal masking ratio for all pre-training configurations on MIMIC-III. We fine-tune the pre-trained model with a label ratio of 10% on the first fold of the MIMIC-III dataset and report validation AUC results in the LOS task. For computational reasons we performed this experiment only over one fold, but as MIMIC-III is a large dataset the findings are general enough as guidance for parameter selection.

Ratio	0.15	0.3	0.5	0.75	1.0
BM	71.48	72.37	74.30	73.92	**74.46**
TFM	74.29	**75.33**	74.86	–	–
Ratio	0.05	0.1	0.2	0.3	0.4
PM	74.84	**76.83**	75.65	75.74	75.82

Table 6
TADPOLE dataset statistics.

Task	No. of samples	No. of feature	No. of features for graph	No. of classes
Disease Pred.	564	12	12	3

Table 7
TADPOLE class occurrences.

CN	MCI	AD
28.55%	56.56%	14.89%

4.1.2. TADPOLE

Additionally to MIMIC-III, which is a classical EHR dataset, we evaluate our method on the TADPOLE dataset (Marinescu et al., 2018), extending our method to multi-modal clinical data. Moreover, this allows us to test our method on a less complex dataset including only static data. This data was obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). The ADNI was launched in 2003 as a public–private partnership, led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI has been to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease (AD). TADPOLE contains a subset of ADNI comprising data from visits of 564 patients. For each patient, we use a restricted set of 12 features, which were claimed to be informative by the TADPOLE challenge. They include demographics (age, gender, and occurrence of apoe4 gene), results of four cognitive tests, and five imaging features, which are extracted from MR and PET imaging. While the demographics and cognitive test results are discrete, the imaging features are continuous. We normalize the continuous features between zero and one. As a downstream task, we perform diagnosis prediction, classifying each patient into one of the following groups: Cognitive Normal (CN), Mild Cognitive Impairment (MCI), or Alzheimer's Disease (AD). We only use data from patients' first visits to avoid leakage of information. Table 6 provides some basic statistics about the TADPOLE dataset and Table 7 the class occurrences for disease prediction.

Graph Construction: To construct a patient population graph on TADPOLE, we compute a feature similarity between all patients. As we work with a reduced feature set of only 12 features, we decided to use all features to compute the feature similarity. Dependent on the feature type this similarity is computed differently. For the demographics, age, gender, and apoe4, we use absolute feature matching, checking if patients have the same gender/apoe4 feature or are of similar age:

$$S_{dem}(r_i, r_j) = \sum \begin{cases} 1 \text{ if } f_i = f_j \text{ else } 0 \\ 1 \text{ if } \left|age_i - age_j\right| \leq 2 \text{ else } 0 \end{cases} \div 3, \tag{5}$$

where f = (apoe4, gender).

Table 8
Experiments to find the optimal masking ratio for pre-training on TADPOLE. We fine-tune with a label ratio of 1% and evaluate with 10-fold cross-validation.

Ratio	0.1	0.2	0.3	0.4
AUC	91.45 ± 3.11	93.07 ± 2.29	**93.49 ± 2.07**	93.37 ± 1.95

Table 9
Sepsis Prediction dataset statistics.

Task	No. of samples	No. of features	No. of classes	Occ. Sepsis
Sepsis Pred.	40 336	40	2	7.27%

For the cognitive test results $\mathbf{d_i}$, which are discrete but ordinal features, and the continuous imaging features $\mathbf{c_i}$, we calculate the respective normalized L2 distances:

$$S_{cog}(r_i, r_j) = \frac{\sum_{f \in \mathbf{d_i}} \|f_{r_i} - f_{r_j}\|}{max(\mathbf{d_i})} \tag{6}$$

$$S_{img}(r_i, r_j) = sig(\sum_{f \in \mathbf{c_i}} \|f_{r_i} - f_{r_j}\|). \tag{7}$$

Given all these similarities, we construct a k-NN graph with k = 5, dependent on the mean similarity (S), which is computed as the mean of S_{dem}, S_{cog} and S_{img}. This leads to on average 6.1 ± 1.1 separated components in the graph.

Pre-Training Configuration: As TADPOLE is a less complex dataset and does not include time-series data, we apply only Static Feature Masking as a pre-training task. For this, we apply masking to the APOE4 gene feature, the cognitive test results, and the imaging features with a masking ratio of 30%. We set the masking ratio experimentally, as shown in Table 8. Although Static Feature Masking is a simple masking technique, this experiment allows us to evaluate the benefit of our overall framework on static, multi-modal, clinical data, including the modeling as a patient population graph and the combination of masking-based pre-training with a transformer-based architecture.

4.1.3. Sepsis prediction dataset

We work with the Sepsis Prediction dataset from the PhysioNet/Computing in Cardiology Challenge 2019 (Reyna et al., 2019; Goldberger et al., 2000). It includes longitudinal EHR data, including vital signs and laboratory values, from patient stays in two different hospitals, as well as the patient's demographics.

The publicly available part of the Sepsis Prediction dataset includes 40,336 ICU patients, equally distributed between two hospitals. The features of every patient include eight vital signs, 26 laboratory values, and six demographics. As for MIMIC-III, the measurement features (vital signs and laboratory values) are time-series features with hourly granularity and a high ratio of missing values. Table 9 provides some basic statistics about the Sepsis Prediction dataset.

For every patient, an hour-wise label indicates if the patient has sepsis. In this work, we target a patient-level binary classification instead of an hourly prediction and predict if a patient will develop sepsis or not. We crop the data of each patient such that for septic patients we only see a time window of 24 h ending six hours before the onset of the sepsis. For non-septic patients, we select a random window of 24 h. Thus we still task the model to identify sepsis before its onset, but we concentrate on the time window close to onset. Further we apply the same data pre-processing as in the MIMIC-III dataset, including the normalization of continuous features to the normal distribution and interpolation of missing values.

The graphs for the Sepsis Prediction dataset are created analogously to MIMIC-III. Here this results in 17.7 ± 4.1 components per graph.

Comparison to MIMIC-III Like MIMIC-III, the Sepsis Prediction dataset contains longitudinal measurement features and patient demographics. For the demographics, both datasets contain age and gender, the other demographics are different. From the measurements in the

Table 10
Learning rates for pre-training on MIMIC-III.

	TFM	BM	TP	PM	MT
lr	1e−3–1e−4	1e−3–1e−4	5e−4	5e−4	5e−4

Table 11
Learning rates for from scratch training (SC) and fine-tuning (FT) on the MIMIC-III dataset.

Task	LOS		ACU		ACU-4	
	SC	FT	SC	FT	SC	FT
lr	1e−4	1e−5	1e−4	1e−5	1e−4	1e−5/5e−5[a]

[a]5e−5 when pre-training with Patient Masking, else 1e−5.

Sepsis Prediction dataset, 24 features are also included in MIMIC-III, while 10 features are new. Further, MIMIC-III contains additional measurement features that are not included in the Sepsis Prediction dataset. Moreover, the Sepsis Prediction dataset does not contain any treatment features.

4.2. Implementation details

All experiments are implemented in PyTorch and performed on a TITAN Xp GPU with 12 GB VRAM. For cross-validation, pre-training is performed separately on the training data of every fold to avoid leaking information from the validation or test sets during pre-training. All experiments are optimized using the Adam optimizer (Kingma and Ba, 2014). We manually tuned hyperparameters per dataset separately for each pre-training task and for from scratch training as well as for fine-tuning of each downstream task.

MIMIC-III: The backbone for the MIMIC-III model consists of $\mathbf{L} = 8$ Graphormer layers. The best model is selected given the performance on the validation set, and all models are trained until convergence of the validation performance. For a fair comparison with the state of the art, all results are averaged over six folds as provided by McDermott et al. (2021), each with an 80-10-10 split into train, validation, and test data, and computed over the respective test sets. The learning rates for the different pre-training and fine-tuning tasks on MIMIC-III are given in Tables 10 and 11. In Table 10 we denote the use of a polynomial decaying learning rate as $start\_lr - end\_lr$.

TADPOLE: The backbone for the TADPOLE model consists of $\mathbf{L} = 4$ Graphormer layers. For pre-training, we train the model with a learning rate of 1e–5 for 6000 epochs. For from scratch training on the downstream task, we use a polynomial decaying learning rate, which is reduced from 1e–5 to 5e–6 for 1200 epochs. To fine-tune pre-trained models, we use a learning rate of 5e–6 and perform training again for 1200 epochs. When training with a label ratio of 1% we reduce the epochs to 200, as for this small amount of data, the pre-trained models reach optimal performance much faster than other models. All results are computed using 10-fold cross-validation.

Sepsis Prediction dataset: We use the same model as for the MIMIC-III dataset with $\mathbf{L} = 8$ Graphormer layers. The encoder is adapted to fit the Sepsis Prediction dataset's features. We perform 5-fold cross-validation, where every rotation is split into train, validation, and test sets with an 80-10-10 split, and select the best model based on the validation set performance per split. We use a learning rate of 1e–4 for both from scratch training and transfer learning.

4.3. Results and discussion

In our experiments, we aim to evaluate the proposed model on different downstream tasks, compare the performance of our model to related work and evaluate the performance improvement that can be reached with our pre-training methods. Further, we compare different pre-training methods to one another and provide various ablation studies, investigating the importance of the different modules of our proposed model architecture.

Table 12
Accuracy and AUC of the proposed method compared with DGM on TADPOLE and the model proposed by McDermott et al. (2021) on MIMIC-III.

Model	ACC	AUC
MIMIC-III: LOS		
Wang et al. (2020) - RF	68.3	73.3
McDermott et al. (2021)	Not reported	71.00 ± 1.00
Proposed	**70.29 ± 1.10**	**76.17 ± 1.02**
MIMIC-III: ACU		
McDermott et al. (2021)	Not reported	78.00 ± 2.00
Proposed	43.67 ± 0.83	**79.42 ± 2.72**
TADPOLE		
Cosmo et al. (2020)	**92.91 ± 02.50**	94.49 ± 03.70
Kazi et al. (2022)	91.05 ± 5.93	96.86 ± 1.81
Proposed	92.59 ± 3.64	**96.96 ± 2.32**

4.3.1. Comparative methods

To show the positive impact of our (Graph)-Transformer-based architecture, we compare our model to related work without any pre-training. Therefore we train our model from scratch on the full training data of MIMIC-III and TADPOLE for different downstream tasks and compare the results to previous work. The results are shown in Table 12.

For MIMIC-III, both targeted downstream tasks, LOS and ACU, were defined in the EHR pre-training benchmark by McDermott et al. (2021). Similar tasks were targeted by previous work, but only a few works target the same tasks on the MIMIC-III dataset. As we use the exact task definitions by McDermott et al. and use the pre-processed dataset they provide, comparing with their work is most meaningful. Further, we compare to MIMIC-Extract (Wang et al., 2020) for LOS. They compared different models for 3-day length-of-stay prediction. We report the performance of the best-performing model, which is a random forest classifier. ACU prediction was so far only targeted by McDermott et al. and our newly defined ACU-4 task was not targeted before, which is why we cannot provide results of previous work. We outperform the random forest model by Wang et al. in LOS as well as the GRU-based model by McDermott et al. in LOS and ACU prediction, showing the benefit of modeling MIMIC-III as a population graph as well as our model architecture (Table 12).

For the TADPOLE dataset, the most important previous works and state-of-the-art models in diagnosis prediction on TADPOLE are DGM (Kazi et al., 2022) and a latent graph learning framework proposed by Cosmo et al. (2020). Similar to our work, they model the data as a population graph, but instead of pre-defining this graph, the proposed models learn a population graph in an end-to-end manner for a given downstream task. This allows the model to decide which patients should be connected depending on the current task. Even though our population graph is fixed, our model still has the freedom to weigh the importance of other patients by learning global attention toward all nodes in the graph. Besides, one recent arxiv paper (Kazi et al., 2021) could further improve disease classification performance on TADPOLE by learning the importance of the given input features. However, it is out of context for this work. We outperform the work by Kazi et al. (2022) and reach similar accuracy as Cosmo et al. (2020) while outperforming in AUC, which is the more important metric as TADPOLE is an imbalanced dataset (Table 12).

Overall, we show significant performance gains on the MIMIC-III dataset and achieve comparable results on TADPOLE. The good results on both MIMIC-III and TADPOLE show that the proposed data modeling and model architecture are a good fit for the task at hand.

Table 13
Performance of the proposed model in accuracy and AUC trained from scratch (SC) or fine-tuned after pre-training (FT) with the different proposed pre-training tasks (TP, BM, TFM, PM, MT) for different label ratios for the LOS task.

MIMIC-III: LOS							
Labels	Metric	SC	FT: TP	FT: BM	FT: TFM	FT: PM	FT: MT
1%	ACC	59.86 ± 2.11	64.40 ± 1.17	63.22 ± 2.39	65.25 ± 1.09	65.72 ± 1.53	**66.39 ± 1.34**
	AUC	62.98 ± 2.55	68.23 ± 1.18	68.07 ± 1.80	69.90 ± 1.26	70.84 ± 2.59	**71.40 ± 1.81**
5%	ACC	64.79 ± 1.16	66.23 ± 0.88	66.82 ± 0.89	68.66 ± 0.73	68.41 ± 0.92	**68.97 ± 0.46**
	AUC	68.85 ± 1.53	70.99 ± 0.03	72.27 ± 1.19	73.97 ± 1.28	74.31 ± 1.04	**74.99 ± 1.00**
10%	ACC	64.72 ± 0.45	66.92 ± 1.10	67.71 ± 0.69	69.42 ± 1.23	69.19 ± 0.58	**69.99 ± 0.97**
	AUC	68.97 ± 0.66	71.57 ± 0.99	73.55 ± 0.60	75.09 ± 1.29	74.92 ± 0.87	**75.90 ± 1.35**
50%	ACC	67.41 ± 1.31	68.87 ± 0.89	69.98 ± 0.69	70.85 ± 0.92	70.71 ± 1.01	**71.46 ± 0.91**
	AUC	72.53 ± 1.08	74.21 ± 1.11	76.02 ± 0.87	76.86 ± 1.47	77.03 ± 0.95	**77.77 ± 1.17**
100%	ACC	70.29 ± 1.10	69.84 ± 1.11	70.73 ± 0.70	71.44 ± 1.25	71.02 ± 0.73	**72.13 ± 1.46**
	AUC	76.17 ± 1.02	75.47 ± 0.86	76.20 ± 0.54	77.78 ± 1.31	77.99 ± 0.90	**78.73 ± 1.21**

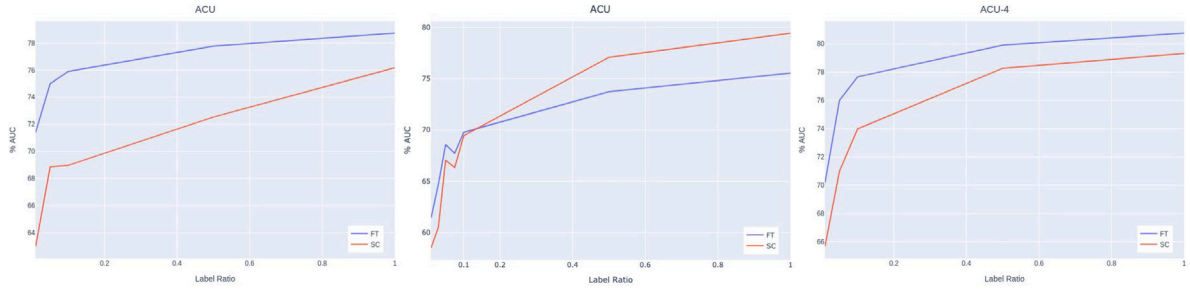

Fig. 3. Visualization of the AUC development comparing from scratch and multi-task pre-training on all tasks on MIMIC-III. The improvement decreases with higher label ratios. For the ACU task pre-training only improves performance for small label ratios up to 10%.

4.3.2. *Effect of pre-training*

The focus of our work lies in investigating the benefit of pre-training for patient-level prediction tasks, especially for scenarios with limited labeled data. To evaluate our proposed pre-training techniques, we compare the performance of our model trained from scratch to our pre-trained and subsequently fine-tuned model. In general, pre-training is especially suitable to be applied in scenarios where much more data is available for pre-training than for fine-tuning. To simulate this case, we artificially create scenarios with limited labeled data, by fine-tuning the model only with a subset of the given labels. We vary the label ratio, meaning the number of labels used for fine-tuning, between 1%, 5%, 10%, 50%, and 100%. On the other hand, we always use 100% of the given data for unsupervised pre-training. The results emphasize the benefits of our unsupervised pre-training with limited labels.

MIMIC-III is a complex dataset, including time-series features and multiple challenging downstream tasks. This allows us to evaluate and compare all of our proposed pre-training strategies on multiple prediction tasks.

For LOS the results are shown in Table 13. Further, the development of the improvement through pre-training for different label ratios is visually shown in Fig. 3 for all tasks on MIMIC-III. For LOS, both accuracy and AUC improve significantly for all label ratios compared to from scratch training, independent of the used pre-training task. When comparing the different pre-training tasks (TM, BM, TFM, and PM), it can be observed that Feature and Patient Masking consistently outperform Block-wise Masking and Treatment Prediction. Further Treatment Prediction overall shows the least positive effect. Considering the difficulty of the pre-training tasks, the easier tasks tend to be less beneficial than the harder ones. Treatment Prediction only asks the model to predict binary indicators if a certain treatment was applied, while the Feature Masking task includes predicting time-series features describing the treatment application in each hour. So conceptually Treatment Prediction is a sub-task of Feature Masking. Similarly during Block-wise Masking the model has to predict feature values only for a few hours while having access to past and future measurements, which makes the task simpler compared to predicting a fully masked feature as in Feature Masking. These results indicate that a certain level of complexity is beneficial for a valuable pre-training task.

For acuity (ACU) prediction we consider both the original as well as our adapted ACU-4 task. However, during our experiments we observed that many of the 18 classes from the original task are not predicted at all, neither when training from scratch nor when fine-tuning. This supports our claim, that the task is not well defined including missing, very rare, and very similar classes. Therefore, we introduce our newly defined ACU-4 task to get more meaningful results.

For ACU prediction, pre-training improves both metrics for small label ratios (1% till 10%) as shown in Table 14. For higher label ratios only accuracy improves. Again Feature Masking consistently outperforms Block-wise Masking and Treatment prediction shows the least performance improvements. When training on the adapted ACU-4 task, a clear benefit of all pre-training methods can be observed (see Table 15). Similar to the LOS task, Treatment Prediction is the least beneficial and Feature Masking is the most beneficial, supporting our hypothesis that the difficulty of the pre-training tasks influences their effect. However, for both acuity prediction tasks (ACU and ACU-4) Patient Masking shows less benefit than Block-wise and Feature Masking, indicating that the information of other patients is less relevant for acuity prediction than for length-of-stay prediction. Generally, Patient Masking is a complex task, but it requires the information of other patients to be helpful in solving the end task. Therefore the value of Patient Masking is dependent on the correlation between different patients, which can differ depending on the targeted task. This also shows that the selection of the optimal pre-training task depends on the current downstream task.

Multi-Task Pre-training further improves the performance after fine-tuning for all downstream tasks, LOS, ACU, and ACU-4. This indicates that the different tasks convey at least partially complementary knowledge and together lead to an improved understanding of the data. Moreover, as described previously the effectiveness of the single pre-training tasks differs between tasks, showing that the selection of the best pre-training task is dependent on the downstream task. With Multi-Task Pre-training this decision can be circumvented, as it generally

Table 14
Performance of the proposed model in accuracy and AUC trained from scratch (SC) or fine-tuned after pre-training (FT) with the different proposed pre-training tasks (TP, BM, TFM, PM, MT) for different label ratios on the ACU task.

MIMIC-III: ACU							
Labels	Metric	SC	FT:TP	FT:PM	FT:BM	FT:TFM	FT:MT
1%	ACC	28.10 ± 2.94	30.019 ± 4.30	32.25 ± 1.59	33.31 ± 2.19	**34.30 ± 2.73**	33.17 ± 1.96
	AUC	58.51 ± 0.80	59.39 ± 2.41	60.86 ± 0.94	59.59 ± 2.70	60.10 ± 2.09	**61.43 ± 1.86**
5%	ACC	35.16 ± 1.88	36.76 ± 1.04	39.06 ± 1.20	39.77 ± 1.05	40.57 ± 0.97	**40.86 ± 0.79**
	AUC	67.06 ± 2.84	63.22 ± 2.23	66.56 ± 1.33	66.93 ± 2.25	67.55 ± 1.58	**68.58 ± 1.27**
10%	ACC	36.69 ± 1.06	36.86 ± 3.49	40.71 ± 0.75	41.93 ± 0.95	**42.47 ± 1.01**	41.91 ± 1.15
	AUC	69.44 ± 2.55	64.28 ± 3.08	66.43 ± 1.86	68.38 ± 2.35	69.13 ± 0.91	**69.77 ± 1.98**
50%	ACC	42.68 ± 1.57	43.52 ± 1.20	42.98 ± 1.29	44.81 ± 0.23	44.87 ± 0.98	**45.39 ± 0.71**
	AUC	**77.07 ± 2.22**	72.30 ± 2.09	68.88 ± 3.15	72.35 ± 3.95	73.78 ± 1.83	73.72 ± 2.39
100%	ACC	43.67 ± 0.83	44.71 ± 1.06	43.78 ± 1.34	45.47 ± 0.58	**46.27 ± 0.76**	46.26 ± 0.85
	AUC	**79.42 ± 2.72**	73.75 ± 2.00	70.66 ± 2.60	75.60 ± 3.48	75.42 ± 1.66	75.54 ± 2.35

Table 15
Performance of the proposed model in accuracy and AUC trained from scratch (SC) or fine-tuned after pre-training (FT) with the different proposed pre-training tasks (TP, BM, TFM, PM, MT) for different label ratios for the ACU-4 task. The best and second-best results per label ratio of all single-task pre-training methods and from-scratch training are marked as bold/green.

MIMIC-III: ACU-4							
Labels	Metric	SC	FT: TP	FT: PM	FT: BM	FT: TFM	FT: MT
1%	ACC	41.03 ± 2.86	43.10 ± 2.61	45.16 ± 2.58	44.78 ± 3.14	46.23 ± 2.79	**46.24 ± 2.31**
	AUC	65.67 ± 1.95	65.70 ± 1.27	68.13 ± 1.76	68.40 ± 3.07	**70.21 ± 2.17**	70.16 ± 1.97
5%	ACC	46.33 ± 2.05	48.10 ± 0.71	50.49 ± 1.27	50.71 ± 1.16	51.77 ± 0.63	**52.68 ± 1.51**
	AUC	71.00 ± 1.17	70.84 ± 0.95	73.59 ± 1.09	75.09 ± 0.65	75.62 ± 0.72	**76.01 ± 1.10**
10%	ACC	49.93 ± 1.02	50.71 ± 0.82	51.63 ± 0.75	52.82 ± 1.48	53.77 ± 1.27	**54.41 ± 1.73**
	AUC	73.99 ± 0.64	73.07 ± 0.28	75.17 ± 1.06	76.90 ± 0.91	77.09 ± 0.71	**77.67 ± 0.79**
50%	ACC	54.02 ± 1.27	54.42 ± 1.16	53.76 ± 1.40	55.93 ± 1.28	56.17 ± 1.18	**56.30 ± 1.27**
	AUC	78.28 ± 0.56	77.69 ± 0.65	77.70 ± 0.98	79.63 ± 0.63	79.66 ± 0.60	**79.92 ± 0.70**
100%	ACC	55.41 ± 0.87	54.98 ± 1.20	54.75 ± 1.03	56.86 ± 1.26	56.84 ± 0.71	**57.11 ± 1.19**
	AUC	79.33 ± 0.72	78.91 ± 0.76	78.72 ± 0.90	80.44 ± 0.67	80.51 ± 0.45	**80.77 ± 0.78**

Table 16
Performance of the proposed model in accuracy and AUC trained from scratch (SC) or fine-tuned after pre-training (FT) for different label ratios on the TADPOLE dataset.

TADPOLE			
Labels	Metric	SC	FT
1%	ACC	59.42 ± 8.40	**78.89 ± 2.45**
	AUC	68.72 ± 12.74	**93.49 ± 2.07**
5%	ACC	78.23 ± 6.83	**83.37 ± 6.29**
	AUC	87.23 ± 4.91	**94.99 ± 2.55**
10%	ACC	87.00 ± 4.86	**87.71 ± 4.65**
	AUC	92.03 ± 3.39	**95.96 ± 2.51**
50%	ACC	**92.41 ± 3.69**	91.52 ± 3.76
	AUC	96.06 ± 2.48	**97.23 ± 1.94**
100%	ACC	**92.59 ± 3.64**	92.24 ± 3.47
	AUC	96.96 ± 2.23	**97.52 ± 1.67**

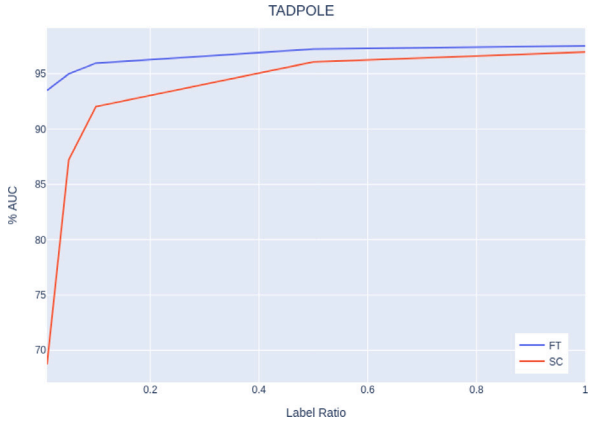

Fig. 4. Visualization of the AUC development comparing from scratch and pre-training on TADPOLE. Again the improvement decreases with higher label ratios.

outperforms the best of the single methods. Consequently, Multi-Task Pre-training is well suited as an out-of-the-box pre-training method, as it is less dependent on the downstream task.

TADPOLE Additionally, we use the less complex TADPOLE dataset to evaluate our method on multi-modal, but static clinical data. The results of the fine-tuned model, which was pre-training with Static Feature Masking, as well as the model trained from scratch for disease prediction are shown in Table 16 and the performance development in relation to the label ratio is visualized in Fig. 4. Even though TADPOLE is a rather small dataset and thus also the amount of data used for pre-training is limited, we show a significant benefit through pre-training. Especially in settings with limited labels (1%, 5%, 10%) pre-training helps to improve performance significantly. While the model trained from scratch starts to overfit very fast when trained with very few samples (e.g. around 50 samples for a label ratio of 1%), the pre-trained model reaches a quite good performance, by quickly adapting to the provided samples. Moreover, the AUC continues to improve when pre-training the model up to the full dataset size, showing that pre-training helps to deal with the class imbalance in this dataset. This experiment shows the value of our method for static, multimodal clinical data, and the benefit of pre-training when labeled data is very limited.

When analyzing the confusion matrices for models trained from scratch and pre-trained models, we can conclude that the major improvement through pre-training in binary tasks (LOS) lies in reducing false negatives. For multi-class tasks such as on TADPOLE or ACU-4 prediction on MIMIC-III, false negatives are mainly reduced for the minority classes while for the majority class false positives are reduced. Overall, this shows that pre-training helps the model to miss fewer positive cases, especially for minority classes.

Table 17
Comparison to other pre-training methods on MIMIC-III. *MI and MT are the masked imputation and multi-task pre-training proposed by McDermott et al. (2021). MI is an unsupervised pre-training method, while MT is supervised.

LOS				
Labels	Metric	MI*	MT*	Proposed
1%	AUC	62 ± 3	67 ± 3	**71.40 ± 1.81**
	gain	+4	**+9**	+8.42
10%	AUC	60 ± 2	65 ± 3	**75.90 ± 1.35**
	gain	−6	−1	**+6.93**
100%	AUC	58 ± 3	64 ± 4	**78.73 ± 1.21**
	gain	−13	64 ± −7	**+2.56**
ACU				
1%	AUC	60 ± 4	**60 ± 1**	**60.10 ± 2.09**
	gain	+1	+1	**+1.59**
10%	AUC	69 ± 3	**70 ± 4**	69.13 ± 0.91
	gain	+0	**+1**	−0.31
100%	AUC	74 ± 2	74 ± 4	**75.42 ± 1.66**
	gain	**−4.0**	**−4.0**	**−4.0**

Table 18
Performance of the proposed model in accuracy, AUC, and F1 score trained from scratch (SC) or fine-tuned after pre-training on MIMIC-III (FT) for different label ratios on the Sepsis prediction task.

Sepsis			
Labels	Metric	SC	FT
1%	ACC	**89.32 ± 1.23**	88.95 ± 2.01
	AUC	68.18 ± 4.29	**70.25 ± 3.01**
	F1	56.32 ± 1.44	**58.28 ± 1.40**
5%	ACC	**89.91 ± 1.12**	89.84 ± 0.49
	AUC	75.60 ± 4.47	**77.16 ± 2.05**
	F1	60.74 ± 1.31	**61.65 ± 1.77**
10%	ACC	**91.12 ± 0.45**	90.86 ± 1.16
	AUC	**80.04 ± 0.86**	79.80 ± 1.26
	F1	62.70 ± 1.98	**64.43 ± 2.35**
50%	ACC	91.32 ± 0.71	**92.47 ± 1.15**
	AUC	80.98 ± 1.77	**83.73 ± 0.93**
	F1	63.95 ± 2.13	**66.94 ± 1.33**
100%	ACC	92.49 ± 0.38	**92.76 ± 0.31**
	AUC	83.40 ± 1.38	**84.49 ± 1.27**
	F1	64.62 ± 1.05	**67.59 ± 1.31**

For the MIMIC-III dataset, McDermott et al. (2021) proposed a benchmark for pre-training on EHR data and evaluated their method among others on Length-of-Stay and Acuity Prediction. Table 17 shows a direct comparison of the pre-training methods proposed by them and our best-performing method (Multi-Task Pre-training). They propose one unsupervised and one supervised pre-training method. In their unsupervised method (MI), they mask and predict random time-points of the EHR record, while for their supervised method (MT), they perform multi-task training on nine out of ten supervised patient outcome prediction tasks and fine-tune the model on the task which was left out. When comparing to the unsupervised masked imputation pre-training our method is clearly superior, both in terms of overall AUC as well as the performance gain achieved by pre-training, showing that our masking strategies better adhere to the nature of this longitudinal and heterogeneous data than simply masking random time points. In comparison to the supervised multi-task pre-training, our method achieves similar performance gains with 1% labels but continues to improve performance in Length-of-Stay prediction also for higher label ratios, while the multi-task pre-training starts to degrade performance. The good results in comparison to this method are especially notable, as we do not use any labels for pre-training, while the supervised pre-training considers labels of multiple prediction tasks. Overall, the results indicate that our modeling of the data as a patient population graph together with our pre-training approaches, which are specifically designed for clinical (time-series) data, are better suited for the given scenario.

Besides the work of Gupta et al. (2020) also propose a method for pre-training on MIMIC-III for phenotype and mortality prediction, but they neither provide an implementation nor sufficient details to re-implement their method from scratch. The results they reported for their unsupervised transfer learning method show only slight improvements in the AUROC score over an LSTM trained from scratch while leading to a slightly worse AUPRC score. For disease prediction on TADPOLE, no previous work proposed a pre-training approach.

Overall we show, that our unsupervised pre-training pipeline helps remarkably to improve performance on both datasets and for multiple patient outcome prediction tasks over patient population graphs. As expected, the highest improvements can be seen in settings with limited labels, however, even when using all labels a notable improvement remains. Further, it can be observed that the pre-trained models mostly have a lower standard deviation than the models trained from scratch, indicating that pre-training also improves model stability. From comparing our different pre-training tasks, we can conclude that the difficulty of the pre-training tasks correlates with their effectiveness.

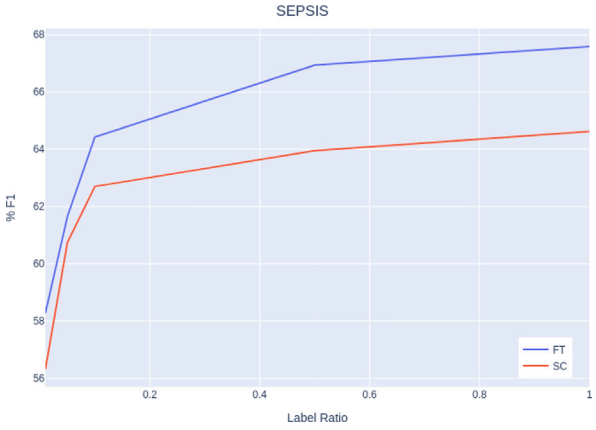

Fig. 5. Visualization of the F1 development comparing from scratch and multi-task pre-training on the Sepsis prediction dataset. Here the performance improvement remains nearly constant for different label ratios.

4.3.3. Application to transfer learning

To evaluate our method in a transfer learning setting, we pre-train our model with multi-task pre-training on MIMIC-III and then fine-tune it on the Sepsis Prediction dataset, which provides overlapping but different features per patient. The results of this experiment are shown in Table 18. As the sepsis prediction task has a very strong class imbalance, we introduce the F1 score as an additional metric. For the sake of completeness, we also report accuracy and AUC such as for the other experiments, but due to the high imbalance, these metrics are less meaningful. The performance development (F1 score) in relation to the label ratio is visualized in Fig. 5. In comparison to the self-supervised settings, where the pre-training effect decreases for higher label ratios, the improvement in transfer learning remains nearly constant for different label ratios. This indicates as we pre-train on a different dataset than we fine-tune, the size difference between the pre-training and fine-tuning set is less relevant. Further, when comparing the from scratch performance to the performance of the pre-trained model, it can be seen that the AUC improves in most cases and the F1 score improves for all label ratios, while accuracy is mostly similar for both models. This shows, that the pre-trained model handles the class imbalance better and improves in predicting the rare positive class by reducing the false negative predictions. This is an important property, as for sepsis prediction it is especially important to classify the positive cases correctly to intervene in time. Overall, even though the MIMIC-III

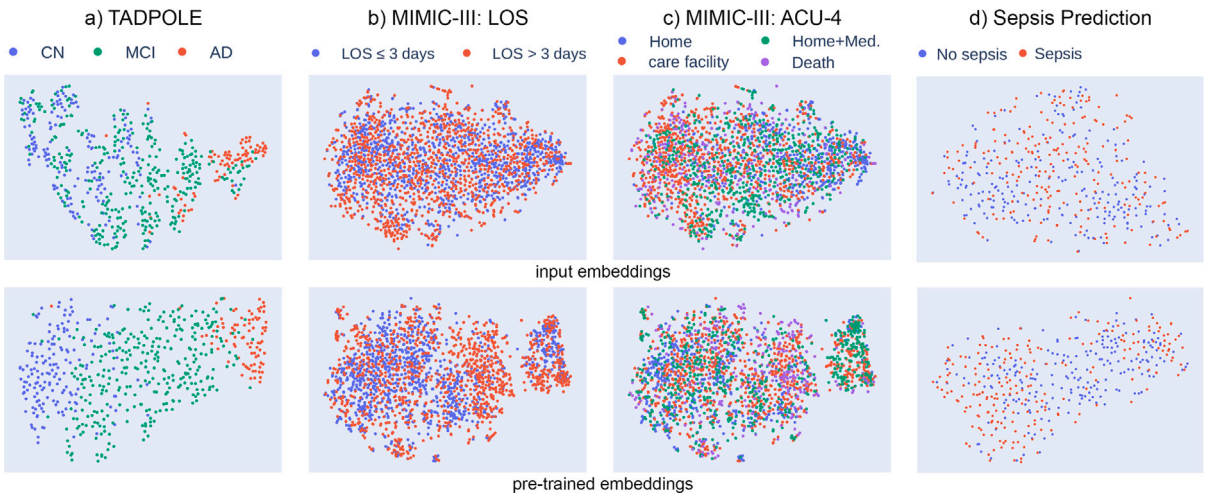

Fig. 6. T-sne visualizations of input and pre-trained embeddings for disease prediction on TADPOLE, LOS, and ACU-4 prediction on MIMIC-III and early prediction of sepsis. The upper row shows the input embeddings and the lower row the pre-trained embeddings. For MIMIC-III we only used the embeddings of five graphs to keep the visualization clear. For the Sepsis Prediction dataset, the model was pre-trained on MIMIC-III with multi-task pre-training.

Table 19
Ablations to test the Graphormer module by replacing it with a linear layer or different GNNs on MIMIC-III (a) downstream task performance trained from scratch (b) results of fine-tuning (FT) on 1% labels, compared to training from scratch (SC) (c) performance in pre-training task.

		Linear	GCN	GAT	GIN	Proposed
a	ACC	67.25 ± 01.11	68.74 ± 01.50	69.22 ± 1.35	68.94 ± 1.11	**70.29 ± 01.10**
	AUC	72.69 ± 00.97	72.64 ± 01.10	74.43 ± 1.10	74.84 ± 1.18	**76.17 ± 01.02**
b	ACC	64.71 ± 0.84	61.77 ± 2.22	62.79 ± 0.91	61.78 ± 1.37	**69.42 ± 1.23**
	gain	+0.93	+1.73	+2.34	+1.10	**+4.70**
	AUC	67.94 ± 1.20	66.03 ± 2.40	66.57 ± 2.03	65.65 ± 1.77	**75.09 ± 1.29**
	gain	+0.22	+3.09	+3.11	+2.13	**+6.12**
c	RMSE	0.838 ± 0.015	0.941 ± 0.027	0.810 ± 0.352	0.951 ± 0.029	**0.783 ± 0.011**
	F1	77.59 ± 0.61	69.97 ± 0.39	81.35 ± 0.51	70.64 ± 0.59	**81.58 ± 0.41**

and the Sepsis Prediction dataset encompass different features, we show that pre-training on MIMIC-III can be used to improve sepsis prediction performance notably, implying that the knowledge acquired during pre-training supports a general understanding of clinical population graphs and is not dataset-specific. This broadens the use case of our method substantially, as it allows pre-training on data with differing features. Thereby, for example, a small hospital with limited access to (labeled) data can use data from larger hospitals for pre-training. Further, it also allows to pre-train with data that was recorded with another application in mind and thus does not exactly match the features of the task at hand.

4.3.4. Effect of embedding space

To give an intuition of why the model initialization learned during pre-training is superior to a random model initialization, we show the effect of pre-training on the distribution of the patient record embeddings (Fig. 6). We compare the embeddings computed by the pre-trained model to the embeddings computed by a randomly initialized model. To visualize these embeddings we use t-sne for dimensionality reduction and visualize the data in a 2D space. Fig. 6 shows the visualizations for both the untrained and the pre-trained models, which are extracted directly before the decoder layer. We used Feature Masking as a pre-training task for these visualizations.

Even though no labels were used for pre-training, the classes are clearly better separated for the pre-trained embeddings. Especially for the TADPOLE dataset, the classes are quite well separated after pre-training (Fig. 6(a)). This explains why for small label ratios, where from scratch training achieves bad results, the effect of pre-training is very high. For the MIMIC-III dataset, pre-training does not lead to a clear class separation, however, in the pre-trained embeddings more clusters of single classes can be seen (Fig. 6(b),(c)). A similar effect can be observed for the Sepsis Prediction dataset. Even though pre-training was performed on MIMIC-III, the septic patients seem to be clustered at the sides of the space, while the non-septic patients are more frequent in the center of the space (Fig. 6(d)). When fine-tuning given this pre-trained initialization, which already clusters nodes with the same class together, the model can easier learn to separate the classes, enabling it to quickly learn a good separation even with little labeled data.

4.3.5. Ablation studies

We perform several ablation studies to evaluate different parts of our proposed model on pre- and downstream task training. All pre-training ablations are performed with limited labels to ensure a well-observable effect in fine-tuning. Specifically, we use a label ratio of 10% on the larger MIMIC-III dataset and of 1% on the TADPOLE dataset. Further, for MIIMC-III we selected one pre-training method and one downstream task for the ablation studies. We perform all ablations using the LOS task and Feature Masking as pre-training, as this pre-training method was overall the best performing one. Further, we analyze the contribution of the different pre-training tasks during Multi-Task Pre-training. We further performed an analysis of the wrongly classified samples, which can be found in the supplementary material.

Effect of Graphormer: We replace the Graphormer module with a linear layer or different graph neural networks (GCN Kipf and Welling, 2016, GAT Veličković et al., 2018 and GIN Xu et al., 2018) and train the model from scratch on the full dataset (see results in Tables 19(a) and 20(a)). The number of layers, the learning rate, and the training epochs are optimized for every conducted ablation.

On the MIMIC-III dataset, the proposed model outperforms the linear model as well as all models based on other GNN architectures, showing that for this more complex dataset using a population graph as well as the use of the attention-based Graphormer layers are beneficial.

Table 20
Ablations to test the Graphormer module by replacing it with a linear layer or different GNNs on TADPOLE (a) downstream task performance trained from scratch (b) results of fine-tuning (FT) on 1% labels, and performance gain compared to training from scratch (c) performance in pre-training task.

		Linear	GCN	GAT	GIN	Proposed
a	ACC	91.14 ± 02.62	74.27 ± 06.41	73.09 ± 5.94	73.47 ± 6.85	**92.59 ± 03.64**
	AUC	**97.77 ± 01.59**	89.89 ± 04.12	89.65 ± 4.39	89.40 ± 4.81	96.96 ± 02.23
b	ACC	73.56 ± 12.30	61.87 ± 5.03	60.58 ± 5.46	59.59 ± 4.80	**74.39 ± 7.58**
	gain	+12.26	+3.67	+0.45	+5.02	**+14.97**
	AUC	**92.33 ± 3.47**	77.68 ± 4.84	74.52 ± 6.08	77.25 ± 5.05	89.98 ± 3.57
	gain	+18.33	+9.62	+3.14	+9.17	**+21.26**
c	RMSE	0.140 ± 0.010	0.140 ± 0.020	0.131 ± 0.018	0.126 ± 0.012	**0.116 ± 0.008**
	ACC	65.33 ± 4.76	**81.71 ± 5.25**	80.78 ± 3.67	80.86 ± 5.12	69.47 ± 6.20

For TADPOLE, the linear model reaches slightly better performance in terms of AUC, while the less complex GNNs reach inferior performance compared to the more complex Graphormer model. The good performance of the linear model shows, that using a population graph is not as beneficial for TADPOLE, which is a relatively small and easy dataset. Instead, each node's features give sufficient information to solve the downstream task. This is also apparent when considering the worse performance of GCN, GAT, and GIN. These models are based on neighborhood aggregation and in each update step the information of only the neighbors can be used, which seems to have a negative effect. Graphormer on the other hand can counteract this effect by using a node-level attention mechanism over the full graph, making it possible to select information from specific nodes in the graph or none at all.

Further, we perform pre-training followed by fine-tuning with limited labels for the different ablations of our model and compare the effect of pre-training the different ablation models with the effect on our proposed model. These results are shown in Tables 19(b) and 20(b). Our proposed unsupervised pre-training method proves to be beneficial for the linear model as well as for the other GNN models, indicating that the pre-training technique is transferable to other models as well. However, the effect of pre-training is highest for the proposed graph transformer based model, showing the value of combining transformer based models with masking-based pre-training tasks. On MIMIC-III we can also see a benefit of using a graph model compared to the linear model in the pre-training effect, showing that for this complex dataset the modeling and processing as a population graph during pre-training has a positive effect.

Tables 19(c) and 20(c) show the masked imputation performance during pre-training, measured by RMSE for continuous features, so measurement features for MIMIC-III and imaging features for TADPOLE, F1 score for the discrete treatment features in MIMIC-III and accuracy for the discrete TADPOLE features (apoe4+cognitive tests). For the cognitive tests in TADPOLE, we use clinically-motivated error margins in which a prediction is considered correct, to compute the accuracy. The used margins are shown in Table 21. The predicted cognitive test scores are ordinal features, so close scores imply a similar performance of the patient. As many possible scores exist, it is a very hard task to make an accurate prediction. By applying error margins, we can assess better, if the model predicts test results in the correct range of values. Overall the results show, that the performance in the pre-training task is not always directly linked to the benefit of pre-training. Especially on TADPOLE, it can be observed, that while GCN, GAT, and GIN perform very well in predicting the cognitive test results, their pre-training benefit is restricted. However, their performance in predicting the imaging features is less than for the proposed model. On MIMIC-III, the proposed Graphormer-based model outperforms all other models in the pre-training task, which is in line with the superior fine-tuning results. In summary, while Graphormer has not always the best performance in the pre-training task, the extracted knowledge is used more beneficially for improving the downstream task, leading to a better fine-tuning performance on both datasets.

Moreover, to better understand the effect of the patient-level attention in the graph transformer module, we visualize the learned attention of the best-performing models for two samples from the TADPOLE and MIMIC-III dataset for Alzheimer's disease and length-of-stay prediction in Fig. 7. We compute the attention by averaging over the layers and heads of the model. It can be observed that the attention to patients close to the current patient is higher than to more distant ones. Moreover, it can be seen that while the current patient is always attended notably, the model also takes other patients into account, showing the model makes use of the information provided in the graph.

Table 21
Clinically-motivated, feature-dependent error margins for measuring cognitive test results prediction performance during pre-training on TADPOLE.

Cognitive test	Number of classes	Error margin
CDRSB	19	4
ADAS11	107	15
MMSE	11	2
RAVLT_immediate	68	5

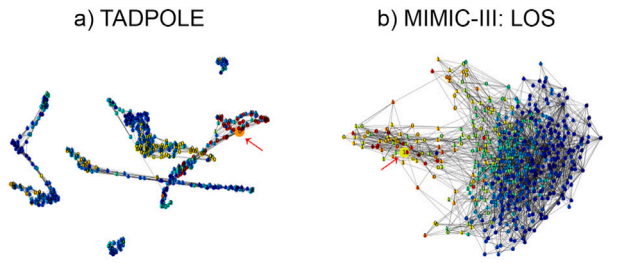

Fig. 7. Visualization of the node-level attention of Graphormer. The used color map displays the highest attention in red, going to yellow, green, and blue for the lowest attention. The current node for which the attention is computed is visualized larger in the image and indicated by the red arrows. The attention is distributed over the whole graph but focuses more on closer nodes.

Effect of Transformer: For MIMIC-III, we insert transformer layers in the encoder to deal with time-series data. To test the effect of these transformer layers, we compare the performance with and without these layers for both from scratch training and fine-tuning. The results of this ablation are shown in Table 22. Part (a) in Table 22 shows the results, when training the proposed model from scratch on the full dataset with and without the transformer layer. This results in a reduction of accuracy and AUC, showing that the transformer layers are helpful for processing the time-series input data of MIMIC-III.

For pre-training on MIMIC-III, the model needs to predict time-dependent outputs, suggesting that here the transformer layers should be especially important, as they can understand the temporal context. To investigate the effect of transformer during pre-training, we remove the transformer layer and compare the effect of pre-training the model when fine-tuning it with limited labels, with and without the transformer layer (see Table 22(b)). Further, we show the performance in the pre-training task for both models (see Table 22(c)). For training from scratch with limited labels, the use of transformer layers improves performance. However, the fine-tuned model without transformer layers

Table 22
All results when removing the transformer layers on MIMIC-III. (a) proposed model on the full dataset without pre-training (b) from scratch training and fine-tuning with 10% labels (c) performance in pre-training task.

		Without transformer	With transformer
a	ACC	69.39 ± 0.60	**70.29 ± 1.10**
	AUC	75.03 ± 1.23	**76.17 ± 1.02**
b	ACC - SC	63.79 ± 0.54	**64.72 ± 0.45**
	ACC - PT	69.34 ± 1.05	**69.42 ± 1.23**
	gain	**+5.55**	+4.70
	AUC - SC	68.41 ± 1.51	**68.97 ± 0.66**
	AUC - PT	**75.19 ± 1.03**	75.09 ± 1.29
	gain	**+6.78**	+6.12
c	RMSE	**0.770 ± 0.016**	0.783 ± 0.011
	F1	80.06 ± 0.49	**81.58 ± 0.41**

reaches a slightly better accuracy and a slightly worse AUC than with the transformer layer. Overall the performance is very similar to the fine-tuned model with transformer layers. Likewise, the performance in the pre-training task of these two models is very similar. This indicates that the pre-trained Graphormer-based model is expressive enough to learn and benefit from the pre-training task and thus less reliant on the transformer layers.

Overall, the transformer layers improve model performance when trained from scratch. However, for pre-training Graphormer, they have little effect, as Graphormer alone already performs well in the pre-training task and benefits from it in fine-tuning.

Multi-task Pre-Training: Our results show that combining different pre-training tasks can help to improve downstream prediction results after fine-tuning even further. In our experiments, we combined all four pre-training tasks we proposed for clinical time-series data as given in the MIMIC-III dataset. To analyze how important each of the four proposed tasks is for multi-task pre-training, we pre-train our model once with all combinations of three pre-training tasks, always leaving out the fourth task. We then fine-tune the models for Length-of-Stay prediction on MIMIC-III and compare the results as shown in Table 23. Our main observation is, that using all four tasks in most cases delivers the best results or is very close to the best result after fine-tuning on the downstream task. This indicates that all four tasks are important to improve the model's understanding of some aspects of the data. Further, leaving our Patient or Feature Masking leads to the overall worst results, while when leaving out Treatment Prediction or Block-wise Masking the performance remains higher. This shows that Patient and Feature Masking have a higher contribution to the overall performance. This aligns with the results from pre-training with single pre-training tasks for LOS, as here these two tasks also lead to the highest benefit for downstream task performance.

Effect of high-dimensional imaging data:

We demonstrated the applicability of our method for fusing imaging and non-imaging data on the TADPOLE dataset, which combines features extracted from MRI and PET with cognitive test results, demographic data and, genetics. To further analyze the capability of our model when using high-dimensional imaging features and showing the promise of fusing multi-modal features we extended our experiments on TADPOLE to a higher number of input features. The increased feature set includes 9 cognitive test results, 7 genetic and demographic features as well as 326 imaging features from three imaging modalities (MRI, PET, DTI), including various biomarkers and some image quality measures. Further, we analyze the performance of our model when restricting the input to only imaging and only non-imaging features. Table 24 shows the results of this experiment. It can be observed that the performance of our model remains very similar when introducing the high-dimensional imaging features, showing the robustness of our model to this kind of data. When using only imaging features the performance drops significantly, and also dropping the imaging features leads to a performance decrease. This shows the importance of the multi-modal feature fusion our method enables.

Furthermore, our general approach can be not only valuable in the analysis of static imaging biomarkers such as in TADPOLE, but also for temporal imaging biomarkers, extracted from dynamic imaging modalities such as dynamic MRI (Pötsch et al., 2021; Zheng et al., 2021), PET (Noortman et al., 2020) or Ultrasound (Ma et al., 2021). Further, it can be applied to longitudinal imaging studies with frequent measurements, such as in the e.g. positioning verification of non-small cell lung cancer, where CT scans are usually acquired daily and their features could be used for treatment adaption (van Timmeren et al., 2017).

We further demonstrate that our model can be adapted to a setting including spatial images. Following Soenksen et al. (2022), who analyzed the use of multi-modal data for patient outcome prediction, we collect fused data from the MIMIC-IV (Johnson et al., 2020) and the MIMIC-CXR (Johnson et al., 2019) datasets. Each patient record includes demographic, measurement, and treatment data very similar as described in section 4.1.1. (MIMIC-III). Further, each record includes one Chest X-ray image of the patient. We use this dataset in order to test the applicability of our model in a setup where numerical EHR information, as well as spatial imaging data, is available. We consider the task of 48 h mortality prediction given the demographics of the patient, an X-ray image at a time-point t, and measurement and treatment data about the last 24 h before time-point t. The task is to predict whether the patient will die within the next 48 h. To integrate the spatial imaging information into our model, we employ a chest X-ray encoder based on a DenseNet backbone, pre-trained for ChestXRay pathology classification (Cohen et al., 2022). The used image encoder has around 6.9 million parameters, of which we fine-tune the last block of 2.2 million parameters. In contrast, the graph encoder has around 5.6 million parameters. Our experiments showed that these encoder sizes were a good trade-off avoiding overfitting but allowing to capture all information in the data. We apply a late fusion approach to fuse the embedding representation given by the X-ray encoder with the output of our data embedding module. For this, we concatenate the two representations before processing them with our decoder.

Table 25 shows the results of this experiment. We observe a substantial performance increase by fusing the imaging and non-imaging data compared to only considering a single data source, demonstrating that our model successfully fuses the multi-modal data and showing the applicability of our model for tasks including spacial imaging data. In the context of pre-training, integrating imaging data needs further investigation, as our pre-training method is primarily designed for understanding heterogeneous but numerical inputs. Therefore, our pretrained EHR encoder may not be sufficient for effectively capturing the relationships between the EHR data and the imaging data. Further, during the pre-training phase, our method did not encounter any spatial imaging data or the combination of EHR and imaging data. As a result, the pre-trained model may not have developed an adequate understanding of the relationships between these two data types, which could have contributed to the lack of improvement in the new setting. We believe it is an important step for applications involving spatial image information to investigate joined pre-training on spacial imaging data and other EHR data, which could be a potential future work.

Overall, our results show the applicability of our model to settings with spatial imaging data and further highlight the importance of considering comprehensive patient data in the field of medical imaging analysis.

5. Conclusion

In this paper, we propose multiple novel unsupervised pre-training methods for multi-modal clinical record data based on masked imputation. We propose to model the clinical data in a patient population graph, such that the model can use other patients' information for

Table 23
Results of multi-task pre-training with all four tasks compared to training with all but one task. The best result per label ratio is bold, while the worst and second-worst results are marked with bold red and red. Fine-tuning on LOS prediction.

MIMIC-III: Multitask Pre-training						
Labels	Metric	MT	MT: no TP	MT: no BM	MT: no TFM	MT: no PM
1%	ACC	**66.39 ± 1.34**	**66.01 ± 0.77**	66.07 ± 1.74	66.37 ± 1.37	66.11 ± 1.12
	AUC	71.40 ± 1.81	**70.57 ± 1.26**	71.08 ± 2.36	**71.45 ± 1.46**	71.42 ± 1.76
5%	ACC	68.97 ± 0.46	68.98 ± 0.98	**69.15 ± 1.07**	**68.59 ± 0.56**	68.71 ± 1.06
	AUC	**74.99 ± 1.00**	74.81 ± 1.08	74.62 ± 1.36	**74.32 ± 1.05**	74.78 ± 1.21
10%	ACC	**69.99 ± 0.97**	69.54 ± 1.05	69.66 ± 0.60	69.10 ± 0.71	**69.03 ± 1.21**
	AUC	**75.90 ± 1.35**	75.53 ± 1.06	75.70 ± 1.10	**74.77 ± 0.48**	75.40 ± 1.09
50%	ACC	**71.46 ± 0.91**	71.10 ± 0.56	71.29 ± 1.12	**70.85 ± 0.88**	71.01 ± 0.92
	AUC	77.77 ± 1.17	**77.79 ± 0.85**	77.58 ± 1.23	77.23 ± 1.15	**76.98 ± 0.91**
100%	ACC	**72.13 ± 1.46**	72.03 ± 0.61	71.53 ± 0.62	**71.34 ± 0.85**	71.61 ± 0.64
	AUC	**78.73 ± 1.21**	78.69 ± 0.91	78.44 ± 1.22	78.27 ± 1.00	**77.88 ± 0.78**

Table 24
Results on TADPOLE dataset with an increased feature set. Further, we analyze the model performance when restricting the input to only imaging or non-imaging features.

	Ours	All features	Imaging	Non-imaging
AUC	96.96 ± 2.13	96.14 ± 3.37	72.74 ± 2.18	95.53 ± 2.33
ACC	92.59 ± 3.64	91.31 ± 4.85	42.90 ± 2.90	89.39 ± 2.56

Table 25
Results on the merged MIMIC-IV and MIMIC-CXR dataset for mortality prediction using only imaging, only non-imaging information or fused information. We further show the effect of our pre-training in this setting.

	Imaging	Non-imaging	Fused	Fused PT
AUC	73.10 ± 6.91	82.95 ± 1.37	**85.87 ± 3.37**	84.50 ± 1.74
ACC	67.90 ± 9.82	74.09 ± 4.73	**78.54 ± 3.79**	75.24 ± 2.99

both pre-training and fine-tuning. For this setup, we present several pre-training tasks, designed to learn a general understanding of multi-modal clinical data. Moreover, we propose a network architecture based on transformer and graph transformer for pre-training and learning on patient population graphs built from heterogeneous clinical data. We show the superiority of the proposed pipeline for various prediction tasks on three datasets and provide an extensive analysis of our method showing different ablation studies. We test our method on MIMIC-III and TADPOLE for self-supervised pre-training and a Sepsis Prediction dataset in a transfer learning setup. All datasets contain medical patient records but encompass different features. We show that pre-training improves results for all datasets over the full dataset size both in the self-supervised as well as the transfer learning scenario. Overall, pre-training proves to be especially helpful for scenarios where only a limited amount of labeled data is used for fine-tuning. Moreover, we compare our different pre-training approaches to one another, showing that more complex tasks tend to have a larger positive impact on the prediction results of downstream tasks. Finally, we combine our proposed pre-training tasks in a multi-task setup, leading to an additional performance gain. Our pre-training methods are all unsupervised and as such task-independent, making them likewise applicable to self-supervised pre-training and transfer learning. Thus the proposed pipeline opens a path to improve learning over multi-modal clinical data on small datasets or datasets with limited labels via pre-training on large unlabeled collections of clinical records from either the same or a different data source.

Declaration of competing interest

We have no conflicts of interest to declare.

Data availability

The link to the code is shared in the manuscript.

Acknowledgments

Anees Kazi's financial support was provided by BigPicture. Data collection and sharing for one of the datasets used this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir, Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Therapeutic Research Institute at the University of Southern California. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of Southern California.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.media.2023.102895.

References

Agrawal, M.N., Lang, H., Offin, M., Gazit, L., Sontag, D., 2022. Leveraging time irreversibility with order-contrastive pre-training. In: International Conference on Artificial Intelligence and Statistics. PMLR, pp. 2330–2353.

Ahmedt-Aristizabal, D., Armin, M.A., Denman, S., Fookes, C., Petersson, L., 2021. Graph-based deep learning for medical diagnosis and analysis: Past, present and future. Sensors 21 (14), http://dx.doi.org/10.3390/s21144758.

Alsentzer, E., Murphy, J.R., Boag, W., Weng, W.-H., Jin, D., Naumann, T., McDermott, M., 2019. Publicly available clinical BERT embeddings. arXiv preprint arXiv:1904.03323.

Bai, W., Chen, C., Tarroni, G., Duan, J., Guitton, F., Petersen, S.E., Guo, Y., Matthews, P.M., Rueckert, D., 2019. Self-supervised learning for cardiac mr image segmentation by anatomical position prediction. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 541–549.

Bao, H., Dong, L., Wei, F., 2021. Beit: Bert pre-training of image transformers. arXiv preprint arXiv:2106.08254.

Bojanowski, P., Joulin, A., 2017. Unsupervised learning by predicting noise. In: International Conference on Machine Learning. PMLR, pp. 517–526.

Caron, M., Bojanowski, P., Joulin, A., Douze, M., 2018. Deep clustering for unsupervised learning of visual features. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 132–149.

Chen, L., Bentley, P., Mori, K., Misawa, K., Fujiwara, M., Rueckert, D., 2019. Self-supervised learning for medical image analysis using image context restoration. Med. Image Anal. 58, 101539.

Cohen, J.P., Viviano, J.D., Bertin, P., Morrison, P., Torabian, P., Guarrera, M., Lungren, M.P., Chaudhari, A., Brooks, R., Hashir, M., et al., 2022. TorchXRayVision: A library of chest X-ray datasets and models. In: International Conference on Medical Imaging with Deep Learning. PMLR, pp. 231–249.

Cosmo, L., Kazi, A., Ahmadi, S.-A., Navab, N., Bronstein, M., 2020. Latent-graph learning for disease prediction. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 643–653.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (Eds.), NAACL-HLT (1). Association for Computational Linguistics.

Ghorbani, M., Kazi, A., Baghshah, M.S., Rabiee, H.R., Navab, N., 2022. Ra-gcn: Graph convolutional network for disease prediction problems with imbalanced data. Med. Image Anal. 75, 102272.

Goldberger, A.L., Amaral, L.A., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.-K., Stanley, H.E., 2000. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. Circulation 101 (23), e215–e220.

Gupta, P., Malhotra, P., Narwariya, J., Vig, L., Shroff, G., 2020. Transfer learning for clinical time series analysis using deep neural networks. J. Healthc. Inf. Res. 4 (2), 112–137.

Han, X., Zhang, Z., Ding, N., Gu, Y., Liu, X., Huo, Y., Qiu, J., Yao, Y., Zhang, A., Zhang, L., et al., 2021. Pre-trained models: Past, present and future. AI Open 2, 225–250.

He, Y., Zhu, Z., Zhang, Y., Chen, Q., Caverlee, J., 2020. Infusing disease knowledge into BERT for health question answering, medical inference and disease name recognition. arXiv preprint arXiv:2010.03746.

Hu, Z., Dong, Y., Wang, K., Chang, K.-W., Sun, Y., 2020b. Gpt-gnn: Generative pre-training of graph neural networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1857–1867.

Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V.S., Leskovec, J., 2020a. Strategies for pre-training graph neural networks. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020.

Huang, Y., Chung, A., 2020. Edge-variational graph convolutional networks for uncertainty-aware disease prediction. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 562–572.

Johnson, A., Bulgarelli, L., Pollard, T., Horng, S., Celi, L.A., Mark, R., 2020. Mimic-iv. PhysioNet, Available Online At: https://Physionet.Org/Content/Mimiciv/1.0/ (Accessed August 23, 2021).

Johnson, A.E., Pollard, T.J., Berkowitz, S.J., Greenbaum, N.R., Lungren, M.P., Deng, C.-y., Mark, R.G., Horng, S., 2019. MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports. Sci. Data 6 (1), 317.

Johnson, A.E., Pollard, T.J., Shen, L., Lehman, L.-w.H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony Celi, L., Mark, R.G., 2016. MIMIC-III, a freely accessible critical care database. Sci. Data 3 (1), 1–9.

Kazi, A., Cosmo, L., Ahmadi, S.-A., Navab, N., Bronstein, M., 2022. Differentiable graph module (dgm) for graph convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell..

Kazi, A., Farghadani, S., Navab, N., 2021. Ia-gcn: Interpretable attention based graph convolutional network for disease prediction. arXiv preprint arXiv:2103.15587.

Kazi, A., Shekarforoush, S., Arvind Krishna, S., Burwinkel, H., Vivar, G., Kortüm, K., Ahmadi, S.-A., Albarqouni, S., Navab, N., 2019a. InceptionGCN: receptive field aware graph convolutional network for disease prediction. In: International Conference on Information Processing in Medical Imaging. Springer, pp. 73–85.

Kazi, A., Shekarforoush, S., Kortuem, K., Albarqouni, S., Navab, N., et al., 2019b. Self-attention equipped graph convolutions for disease prediction. In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019). IEEE, pp. 1896–1899.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kipf, T.N., Welling, M., 2016. Semi-supervised classification with graph convolutional networks. http://dx.doi.org/10.48550/ARXIV.1609.02907.

Komodakis, N., Gidaris, S., 2018. Unsupervised representation learning by predicting image rotations. In: International Conference on Learning Representations (ICLR).

Landi, I., Glicksberg, B.S., Lee, H.-C., Cherng, S., Landi, G., Danieletto, M., Dudley, J.T., Furlanello, C., Miotto, R., 2020. Deep representation learning of electronic health records to unlock patient stratification at scale. NPJ Digit. Med. 3 (1), 1–11.

Li, Y., Rao, S., Solares, J.R.A., Hassaine, A., Ramakrishnan, R., Canoy, D., Zhu, Y., Rahimi, K., Salimi-Khorshidi, G., 2020. BEHRT: transformer for electronic health records. Sci. Rep. 10 (1), 1–12.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., Tang, J., 2021. Self-supervised learning: Generative or contrastive. IEEE Trans. Knowl. Data Eng..

Lu, Y., Jiang, X., Fang, Y., Shi, C., 2021. Learning to Pre-Train Graph Neural Networks. AAAI.

Ma, Q.-P., He, X.-l., Li, K., Wang, J.-f., Zeng, Q.-J., Xu, E.-J., He, X.-q., Li, S.-y., Kun, W., Zheng, R.-Q., et al., 2021. Dynamic contrast-enhanced ultrasound radiomics for hepatocellular carcinoma recurrence prediction after thermal ablation. Mol. Imaging Biol. 23 (4), 572–585.

Malhotra, P., TV, V., Vig, L., Agarwal, P., Shroff, G., 2017. TimeNet: Pre-trained deep recurrent neural network for time series classification. arXiv preprint arXiv:1706.08838.

Marinescu, R.V., Oxtoby, N.P., Young, A.L., Bron, E.E., Toga, A.W., Weiner, M.W., Barkhof, F., Fox, N.C., Klein, S., Alexander, D.C., et al., 2018. TADPOLE challenge: prediction of longitudinal evolution in Alzheimer's disease. arXiv preprint arXiv:1805.03909.

McDermott, M., Nestor, B., Kim, E., Zhang, W., Goldenberg, A., Szolovits, P., Ghassemi, M., 2021. A comprehensive EHR timeseries pre-training benchmark. In: Proceedings of the Conference on Health, Inference, and Learning. pp. 257–278.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. In: Bengio, Y., LeCun, Y. (Eds.), 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings. URL: http://arxiv.org/abs/1301.3781.

Mitani, A.A., Haneuse, S., 2020. Small data challenges of studying rare diseases. JAMA Netw. Open 3 (3).

ml glossary, a., 2022a. Cross-entropy. https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html#cross-entropy, Accessed: 2022-07-04.

ml glossary, b., 2022b. MSE (L2). https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html#mse-l2, Accessed: 2022-07-04.

Noortman, W.A., Vriens, D., Slump, C.H., Bussink, J., Meijer, T.W., de Geus-Oei, L.-F., van Velden, F.H., 2020. Adding the temporal domain to PET radiomic features. PLoS One 15 (9), e0239438.

Ouyang, C., Biffi, C., Chen, C., Kart, T., Qiu, H., Rueckert, D., 2020. Self-supervision with superpixels: Training few-shot medical image segmentation without annotation. In: European Conference on Computer Vision. Springer, pp. 762–780.

Pang, C., Jiang, X., Kalluri, K.S., Spotnitz, M., Chen, R., Perotte, A., Natarajan, K., 2021. CEHR-BERT: Incorporating temporal information from structured EHR data to improve prediction tasks. In: Machine Learning for Health. PMLR, pp. 239–260.

Parisot, S., Ktena, S.I., Ferrante, E., Lee, M., Guerrero, R., Glocker, B., Rueckert, D., 2018. Disease prediction using graph convolutional networks: application to autism spectrum disorder and Alzheimer's disease. Med. Image Anal. 48, 117–130.

Park, S., Bae, S., Kim, J., Kim, T., Choi, E., 2022. Graph-text multi-modal pre-training for medical representation learning. In: Flores, G., Chen, G.H., Pollard, T., Ho, J.C., Naumann, T. (Eds.), Proceedings of the Conference on Health, Inference, and Learning. In: Proceedings of Machine Learning Research, Vol. 174, PMLR, pp. 261–281.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A., 2016. Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2536–2544.

Pennington, J., Socher, R., Manning, C.D., 2014. Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543.

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L., 2018. Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). Association for Computational Linguistics, New Orleans, Louisiana, http://dx.doi.org/10.18653/v1/N18-1202.

Pollard, T.J., Johnson, A.E., Raffa, J.D., Celi, L.A., Mark, R.G., Badawi, O., 2018. The eICU Collaborative Research Database, a freely available multi-center database for critical care research. Sci. Data 5 (1), 1–13.

Pötsch, N., Dietzel, M., Kapetas, P., Clauser, P., Pinker, K., Ellmann, S., Uder, M., Helbich, T., Baltzer, P.A., 2021. An AI classifier derived from 4D radiomics of dynamic contrast-enhanced breast MRI data: potential to avoid unnecessary breast biopsies. Eur. Radiol. 31 (8), 5866–5876.

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., Huang, X., 2020. Pre-trained models for natural language processing: A survey. Sci. China Technol. Sci. 63 (10), 1872–1897.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., 2018. Improving language understanding by generative pre-training.

Rasmy, L., Xiang, Y., Xie, Z., Tao, C., Zhi, D., 2021. Med-BERT: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. NPJ Digit. Med. 4 (1), 1–13.

Reyna, M.A., Josef, C., Seyedi, S., Jeter, R., Shashikumar, S.P., Westover, M.B., Sharma, A., Nemati, S., Clifford, G.D., 2019. Early prediction of sepsis from clinical data: the PhysioNet/Computing in Cardiology Challenge 2019. In: 2019 Computing in Cardiology (CinC). IEEE, pp. Page–1.

Rong, Y., Bian, Y., Xu, T., Xie, W., Wei, Y., Huang, W., Huang, J., 2020. Grover: Self-supervised message passing transformer on large-scale molecular data. Adv. Neural Inf. Process. Syst..

Shang, J., Ma, T., Xiao, C., Sun, J., 2019. Pre-training of graph augmented transformers for medication recommendation. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19. International Joint Conferences on Artificial Intelligence Organization, pp. 5953–5959. http://dx.doi.org/10.24963/ijcai.2019/825.

Soenksen, L.R., Ma, Y., Zeng, C., Boussioux, L., Villalobos Carballo, K., Na, L., Wiberg, H.M., Li, M.L., Fuentes, I., Bertsimas, D., 2022. Integrated multimodal artificial intelligence framework for healthcare applications. NPJ Digit. Med. 5 (1), 149.

van Timmeren, J.E., Leijenaar, R.T., van Elmpt, W., Reymen, B., Lambin, P., 2017. Feature selection methodology for longitudinal cone-beam CT radiomics. Acta Oncol. 56 (11), 1537–1543.

Valenchon, J., Coates, M., 2019. Multiple-graph recurrent graph convolutional neural network architectures for predicting disease outcomes. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 3157–3161. http://dx.doi.org/10.1109/ICASSP.2019.8683433.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y., 2018. Graph attention networks. In: International Conference on Learning Representations.

Verma, S., Zhang, Z.-L., 2019. Learning universal graph neural network embeddings with aid of transfer learning. arXiv preprint arXiv:1909.10086.

Vivar, G., Zwergal, A., Navab, N., Ahmadi, S.-A., 2018. Multi-modal disease classification in incomplete datasets using geometric matrix completion. In: Graphs in Biomedical Image Analysis and Integrating Medical Imaging and Non-Imaging Modalities. Springer, pp. 24–31.

Wang, S., McDermott, M.B., Chauhan, G., Ghassemi, M., Hughes, M.C., Naumann, T., 2020. Mimic-extract: A data extraction, preprocessing, and representation pipeline for mimic-iii. In: Proceedings of the ACM Conference on Health, Inference, and Learning. pp. 222–235.

Wang, X., Yang, S., Zhang, J., Wang, M., Zhang, J., Huang, J., Yang, W., Han, X., 2021. Transpath: Transformer-based self-supervised learning for histopathological image classification. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 186–195.

Wells, B.J., Chagin, K.M., Nowacki, A.S., Kattan, M.W., 2013. Strategies for handling missing data in electronic health record derived data. Egems 1 (3).

Xiao, C., Choi, E., Sun, J., 2018. Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. J. Am. Med. Inform. Assoc. 25 (10), 1419–1428.

Xie, Y., Xu, Z., Zhang, J., Wang, Z., Ji, S., 2022. Self-supervised learning of graph neural networks: A unified review. IEEE Trans. Pattern Anal. Mach. Intell..

Xu, K., Hu, W., Leskovec, J., Jegelka, S., 2018. How powerful are graph neural networks? arXiv preprint arXiv:1810.00826.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V., 2019. XLNet: Generalized autoregressive pretraining for language understanding. In: Advances in Neural Information Processing Systems, Vol. 32. Curran Associates, Inc., URL: https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf.

Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., Liu, T.-Y., 2021. Do transformers really perform badly for graph representation? Adv. Neural Inf. Process. Syst. 34.

Yu, X., Tang, L., Rao, Y., Huang, T., Zhou, J., Lu, J., 2021. Point-BERT: Pre-training 3D point cloud transformers with masked point modeling. arXiv preprint arXiv:2111.14819.

Zebin, T., Rezvy, S., Chaussalet, T.J., 2019. A deep learning approach for length of stay prediction in clinical settings from medical records. In: 2019 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB). IEEE, pp. 1–5.

Zhang, J., Zhang, H., Xia, C., Sun, L., 2020. Graph-bert: Only attention is needed for learning graph representations. arXiv preprint arXiv:2001.05140.

Zheng, R., Shi, C., Wang, C., Shi, N., Qiu, T., Chen, W., Shi, Y., Wang, H., 2021. Imaging-based staging of hepatic fibrosis in patients with hepatitis b: A dynamic radiomics model based on gd-EOB-DTPA-enhanced MRI. Biomolecules 11 (2), 307.