Language Model Training Paradigms for Clinical Feature Embeddings

Yurong Hu[1], Manuel Burger[2], Gunnar Rätsch[2], Rita Kuznetsova[2]
{yurohu, burgerm, raetsch, mkuznetsova}@ethz.ch
[1] Department of Information Technology and Electrical Engineering, ETH Zürich
[2]Department of Computer Science, ETH Zürich

Abstract

In research areas with scarce data, representation learning plays a significant role. This work aims to enhance representation learning for clinical time series by deriving universal embeddings for clinical features, such as heart rate and blood pressure. We use self-supervised training paradigms for language models to learn high-quality clinical feature embeddings, achieving a finer granularity than existing time-step and patient-level representation learning. We visualize the learnt embeddings via unsupervised dimension reduction techniques and observe a high degree of consistency with prior clinical knowledge. We also evaluate the model performance on the MIMIC-III benchmark and demonstrate the effectiveness of using clinical feature embeddings. We publish our code online for replication[1].

1 Introduction

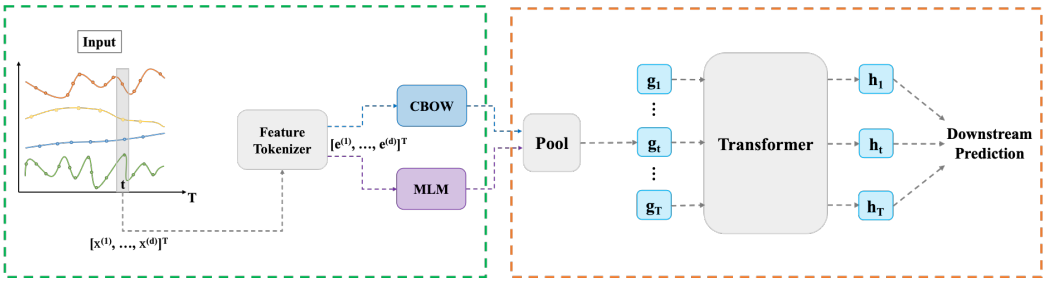
The wide adoption of the EHR system has engendered an unprecedented availability of patient data, which serves as a treasure trove for ML researchers. Such data encapsulates a patient's medical trajectory, inclusive of their medical history, diagnoses, laboratory tests, and treatment interventions. Prior ML research [Horn et al., 2020, Xu et al., 2018] primarily aimed at the modification of the backbone sequence model, mostly employing supervised training approaches for the prediction of patient-centric problems like in-hospital length-of-stay and mortality rates. Simultaneously, several studies [Yue et al., 2022, Tonekaboni et al., 2021, Yèche et al., 2021] have successfully applied self-supervised learning methodologies for the extraction of time-step level or patient-level embeddings in time series. However, these high-level embeddings are largely confined to the specific datasets upon which they were trained. Furthermore, the pre-training objectives are mostly focused on contrastive loss, resulting in a deficit of exploration concerning other predictive objectives. Horn et al. [2020] and Tipirneni and Reddy [2022] considered time series as a set of observation triplets and got the feature level embeddings through the aggregation of three embeddings (i.e., time, feature and value). Both works are confined to the regime of set function learning. Tipirneni and Reddy [2022] also used an auxiliary self-supervision task for training, but we separate the pre-training and fine-tuning stages in our work, which makes it convenient for the unsupervised feature analysis. Other related works are provided in Appendix A.

Our Contribution In this study, we conduct a granular analysis of representation learning for clinical features such as heart rate and blood pressure. We employ self-supervised training paradigms for language models to obtain more universally applicable clinical embeddings. Specifically, we adopt the Continuous Bag of Words (CBOW) model from Word2Vec [Mikolov et al., 2013] and the Masked Language Model (MLM) from BERT [Devlin et al., 2018]. Experimental analysis demonstrates that leveraging clinical feature embeddings can improve the performance on downstream tasks. Additionally, the clinical feature embeddings obtained from the language model pre-training paradigms show a well-structured latent space, from which we can infer established clinical knowledge.

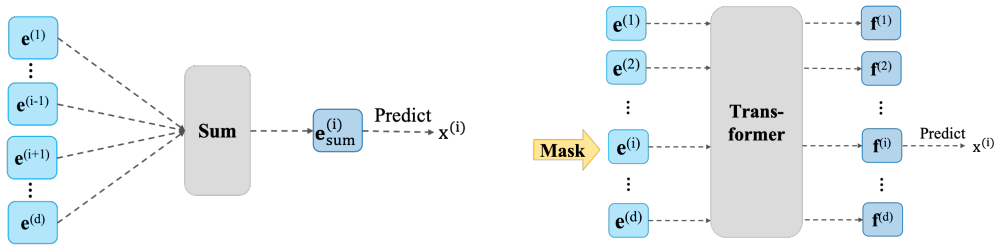
[1]https://github.com/yuroeth/icu_benchmarks

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

2 Methods



(a) **Pipeline of our method.** The green box is the pre-training stage, and the orange box represents the fine-tuning stage.



(b) **CBOW** (c) **MLM**

Figure 1: Self-supervised learning framework for clinical time series.

Notations We define the whole dataset from ICU patient stay as $\{(\mathbf{X}_i, \mathbf{y}_i)|i = 1, 2, ..., N\}$. Each $\mathbf{X}_i$ is a multivariate time series $\mathbf{X}_i = [\mathbf{x}_{i,1}, ..., \mathbf{x}_{i,T}]$, where T is the length of the stay i. Each time step is $\mathbf{x}_{i,t} = [x_{i,t}^{(1)}, ..., x_{i,t}^{(d)}] \in \mathbb{R}^d$, where d is the number of clinical features. Depending on the specific task, the label $\mathbf{y}_i$ for patient stay $\mathbf{X}_i$ can be a single value $\mathbf{y}_i \in \mathbb{R}$ that indicates the state of the whole patient stay or a vector $\mathbf{y}_i \in \mathbb{R}^T$ that corresponds to the state of each time step. In the self-supervised learning stage, we consider each time step $\mathbf{x}_{i,t}$ as one sample for the model. For the ease of expression, we omit the subscript and use $\mathbf{x} = [x^{(1)}, ..., x^{(d)}]$ instead in the following explanation of CBOW and MLM models.

CBOW Given a set of clinical features in a certain time step of patient stay $\mathbf{x} = [x^{(1)}, ..., x^{(d)}]$, we randomly select one numerical variable $x^{(j)}$ and one categorical variable $x^{(k)}$ to predict. The variables are first fed into the feature tokenizer [Gorishniy et al., 2021], see Appendix A, which maps discrete feature values to embedding vectors $\mathbf{e} = [\mathbf{e}^{(1)}, ..., \mathbf{e}^{(d)}]$, where $\mathbf{e}^{(i)} \in \mathbb{R}^m$, and m is the embedding dimension. Then we sum the embeddings of all variables except the predicted one, see Figure 1b: $\mathbf{e}_{sum}^{(j)} = \sum_{i \neq j} \mathbf{e}^{(i)}$ for predicting value $x^{(j)}$ with the mean squared error $L_{num} = \text{MSE}(\text{Linear}(\mathbf{e}_{sum}^{j}), x^{(j)})$ and $\mathbf{e}_{sum}^{(k)} = \sum_{i \neq k} \mathbf{e}^{(i)}$ for predicting value $x^{(k)}$ with the cross-entropy loss $L_{cat} = \text{CE}(\text{Linear}(\mathbf{e}_{sum}^{k}), x^{(k)})$. Finally we add the two losses together $L = L_{num} + L_{cat}$ for the model update.

MLM Based on the initial embeddings $\mathbf{e} = [\mathbf{e}^{(1)}, ..., \mathbf{e}^{(d)}]$ from the Feature Tokenizer [Gorishniy et al., 2021], we randomly mask the embeddings of one numerical variable $\mathbf{e}^{(j)}$ and one categorical variable $\mathbf{e}^{(k)}$, see Figure 1c. Similar to MLM in BERT pre-training [Devlin et al., 2018], we replace the masked positions with (1) the [MASK] embedding 80% of the time (2) a random vector 10% of the time (3) the original feature embedding 10% of the time. The processed embeddings are then passed into the Transformer encoder to get contextual embeddings for each variable $\mathbf{f} = [\mathbf{f}^{(1)}, ..., \mathbf{f}^{(d)}]$. $\mathbf{f}^{(j)}$ and $\mathbf{f}^{(k)}$ are responsible for predicting the corresponding masked variable value $x^{(j)}$ and $x^{(k)}$ respectively. Similarly, we use the total loss $L = L_{num} + L_{cat}$ for model update, where $L_{num} = \text{MSE}(\text{Linear}(\mathbf{f}^{(j)}), x^{(j)})$, $L_{cat} = \text{CE}(\text{Linear}(\mathbf{f}^{(k)}), x^{(k)})$.

Downstream Fine-Tuning For the downstream task, our input is a patient stay comprising several time steps. The learnt embeddings for clinical features at each time step t are pooled to get the time-step level embeddings: $\mathbf{g}_t = \text{Pool}(\mathbf{e}_t^{(1)}, ..., \mathbf{e}_t^{(d)})$ for CBOW and $\mathbf{g}_t = \text{Pool}(\mathbf{f}_t^{(1)}, ..., \mathbf{f}_t^{(d)})$ for

MLM. Then we feed $\mathbf{g}_t$ into the Transformer encoder to get the contextual embeddings for each time step $[\mathbf{h}_1, ..., \mathbf{h}_T] = \text{Transformer}([\mathbf{g}_1, ..., \mathbf{g}_T])$. For time-step level predictions, we apply a linear layer for each time step $\hat{y}_t = \text{Linear}(\mathbf{h}_t)$. For stay level predictions, we apply a linear layer to the last time step counted. Subsequently, we compute the task-specific loss function. In our experiments, we use max pooling to get time-step level embeddings and the cross-entropy loss is adopted for both tasks.

3 Experiment Setup

Dataset We use MIMIC-III dataset [Johnson et al., 2016] for pre-training and fine-tuning. In self-supervised pre-training, we discard time steps with missing value rate larger than 80% (i.e. more than 15 missing values out of 18 features in total). We impute the missing numerical features with the mean and missing categorical features with the mode from the whole dataset. We evaluate the quality of our pre-trained clinical embeddings on two tasks: (1) decompensation and (2) patient mortality at 48 hours after admission from MIMIC-III benchmark [Harutyunyan et al., 2019].

Models We consider two baseline models. The first model is the Transformer [Vaswani et al., 2017] that takes the raw clinical features as input. The second one is the Feature Tokenizer Transformer (FTT) [Gorishniy et al., 2021] which maps the input clinical variables to the embedding vectors before passed to the Transformer encoder. For CBOW and MLM, the feature tokenizer is pre-trained with corresponding self-supervision tasks, as described in Sections 2. The detailed training setup and choice of hyper-parameters are shown in Appendix B.

Metrics Given that the downstream tasks are significantly unbalanced classification problems, we use the area under the precision-recall curve (AUPRC) and the area under the receiver operating characteristics curve (AUROC) as the measurement.

4 Results

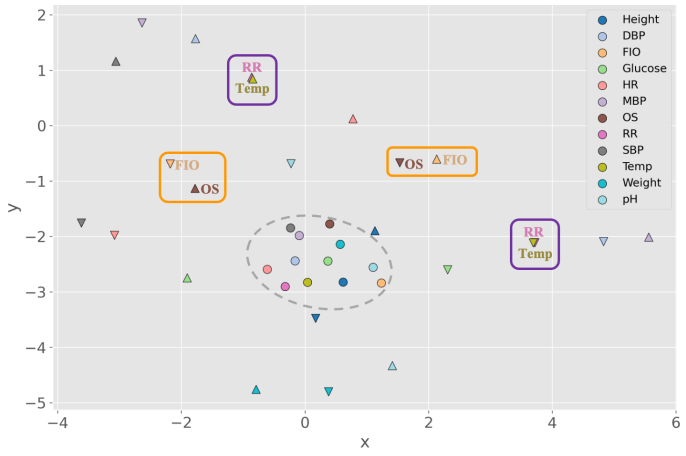
Performance on Downstream Task The pre-trained clinical embeddings are evaluated on the decompensation and mortality prediction tasks from the MIMIC-III benchmark [Harutyunyan et al., 2019]. From Table 1, we see that FTT, CBOW and MLM models outperform the Transformer model, demonstrating that feature embeddings are beneficial to clinical predictions. However, CBOW and MLM can not further improve the performance from the FTT model, which suggests that the pre-training of clinical embeddings does not necessarily help the downstream task. However, from the unsupervised feature analysis below, we will see that the pre-trained embeddings have a better connection with prior clinical knowledge than FTT embeddings.

Task	Decompensation		Mortality	
Metric	AUPRC	AUROC	AUPRC	AUROC
Transformer	34.4 ± 0.4	91.2 ± 0.1	51.5 ± 0.6	$\mathbf{86.5} \pm 0.3$
FTT	$\mathbf{36.4} \pm 0.2$	$\mathbf{91.6} \pm 0.1$	$\mathbf{53.4} \pm 0.4$	85.8 ± 0.1
CBOW	36.3 ± 0.4	91.4 ± 0.1	53.0 ± 0.5	85.8 ± 0.3
MLM	36.2 ± 0.1	$\mathbf{91.6} \pm 0.1$	53.1 ± 0.2	86.0 ± 0.2

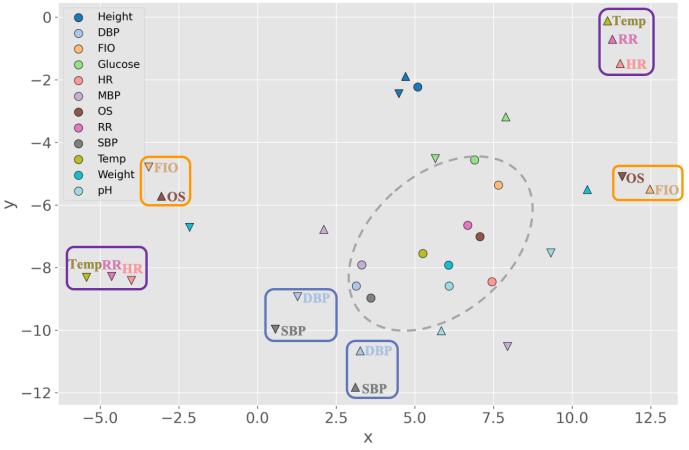
Table 1: Performance on two tasks from the MIMIC-III benchmark for different models measured with AUPRC and AUROC. Mean and standard deviation are reported over three runs.

Unsupervised Feature Analysis For numerical features, artificial feature values are introduced to the pre-trained feature tokenizer, following which the dimension reduction technique T-SNE [Van der Maaten and Hinton, 2008] is employed on the resultant artificial output to enable visualization. Given that the authentic input data is standard normalized (mean $= 0,$ std $= 1$), we choose $(-3 \times \text{std})$, 0, and $(3 \times \text{std})$ as the artificial inputs. Accordingly, low-level feature embeddings are represented as $-3 \times W^{num} + b^{num}$ (▼), middle-level feature embeddings are b^{num} (•), and high-level feature embeddings are $3 \times W^{num} + b^{num}$ (▲). The T-SNE visualizations are depicted in Figure 2. The mapping of feature names to their abbreviations is in Appendix B. On scrutinizing Figure 2, we find that several relationships among pre-trained embeddings align with established clinical knowledge. Both in Figure 2a and Figure 2b, the middle-value feature embeddings ('•') tend to cluster (see gray circles), denoting a normal patient state. Further, we observe a proportional relationship between body temperature and respiratory rate (as well as heart rate in Figure 2b) (see purple rectangles), indicating that an increase in body temperature would correspondingly elevate the respiratory (and heart) rate. Additionally, the MLM embeddings reveal a proportionality between diastolic and systolic blood pressure (see blue rectangles). These correlations are not observed in the embeddings from

FTT, see Appendix C. However, certain inexplicable correlations exist in the pre-trained embeddings, e.g. inversely proportional relationship between FIO and OS. The anticipated correlation would be a direct one, as a higher FIO should theoretically result in a higher OS due to an increase in oxygen absorption into the bloodstream. The discrepancies between the feature clusters and clinical knowledge can be attributed to: (1) the learned embeddings not being impeccable for all features, and (2) the existence of complex interrelationships between different measurements under pathological or extreme conditions. For categorical features, we directly use $W^{cat} + b^{cat}$ as features of varying levels and the results are in Appendix C, where we also present the results for other ablation studies.



(a) **CBOW**



(b) **MLM**

Figure 2: T-SNE visualization, with the perplexity value set to 15, of numerical feature embeddings from CBOW and MLM (FTT can be found in Appendix C). Different colors designate the individual features and shapes their magnitude as explained in Section 4.

5 Conclusion

This work seeks to address the challenges faced in representation learning for clinical time series. Existing works are mainly targeted at learning time-step level or patient level feature representations, with a predominant focus on contrastive losses. While improving predictive performance on downstream tasks, these high-level representations suffer from the confinement to specific datasets they were trained on. In an attempt to improve the universality and applicability of clinical feature representations, our study embarks on a granular analysis of representation learning for clinical features. The primary contributions of our work thus include the derivation of universal clinical feature embeddings via CBOW and MLM models, evaluation and analysis of the pre-trained embeddings, and verification of their effectiveness in performance improvement on downstream tasks and interpetability. We believe our findings will encourage further works in exploring the design and application of clinical embeddings.

References

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

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34:18932–18943, 2021.

Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, Greg Ver Steeg, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific data*, 6(1):96, 2019.

Max Horn, Michael Moor, Christian Bock, Bastian Rieck, and Karsten Borgwardt. Set functions for time series. In *International Conference on Machine Learning*, pages 4353–4363. PMLR, 2020.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-Wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Data descriptor: Mimic-iii, a freely accessible critical care database (2016), 2016.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Sindhu Tipirneni and Chandan K Reddy. Self-supervised transformer for sparse and irregularly sampled multivariate clinical time-series. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(6):1–17, 2022.

Sana Tonekaboni, Danny Eytan, and Anna Goldenberg. Unsupervised representation learning for time series with temporal neighborhood coding. *arXiv preprint arXiv:2106.00750*, 2021.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Yanbo Xu, Siddharth Biswal, Shriprasad R Deshpande, Kevin O Maher, and Jimeng Sun. Raim: Recurrent attentive and intensive model of multimodal patient monitoring data. In *Proceedings of the 24th ACM SIGKDD international conference on Knowledge Discovery & Data Mining*, pages 2565–2573, 2018.

Hugo Yèche, Gideon Dresdner, Francesco Locatello, Matthias Hüser, and Gunnar Rätsch. Neighborhood contrastive learning applied to online patient monitoring. In *International Conference on Machine Learning*, pages 11964–11974. PMLR, 2021.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8980–8987, 2022.

A Related Work

Mikolov et al. [2013] devised two distinctive models that enable the projection of words into a continuous vector space. The Continuous Bag of Words (CBOW) model infers the central word using its surrounding context, whereas the skip-gram model forecasts the neighboring words given the central word. Both models proficiently yield superior quality word representations, effectively capturing both syntactic and semantic word similarities. Subsequent to the advent of the Transformer model [Vaswani et al., 2017], Devlin et al. [2018] introduced BERT, a language representation model developed on the core framework of the Transformer encoder. BERT is engineered to generate deep bidirectional embeddings from a substantial volume of unlabeled text data. Its pre-training tasks include Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). The MLM task prompts the model to predict the masked token within its left and right context, while NSP instructs the model to determine the adjacency of two sentences.

Many studies on time series representation learning focus on obtaining time-step level, stay level or patient level embeddings. Tonekaboni et al. [2021] proposed Temporal Neighborhood Coding (TNC) which leverages the local smoothness inherent to the generative process of time series. They devised a contrastive objective aimed at distinguishing between neighborhood and distant signals. In later work, Yue et al. [2022] proposed TS2Vec, a contrastive learning framework for learning representations of time series data in a hierarchical manner. TS2Vec can apply temporal contrast and instance contrast arbitrarily at each layer of the dilated CNN model. Besides, Yèche et al. [2021] designed a Neighborhood Contrastive Learning (NCL) framework for online patient monitoring. NCL incorporates data augmentation techniques for time series data and a novel contrastive objective, which consists of a Neighbor Alignment objective and a Neighbor Discriminative objective. The patient stay level embeddings learnt in this manner proved effective on the MIMIC benchmark and the Physionet 2019 dataset.

For the acquisition of meaningful representations for clinical features, the initial step is to transform each individual variable into embedding vectors. Gorishniy et al. Gorishniy et al. [2021] summarized the commonly used model architectures for such vectorization, encompassing MLP, ResNet, and Feature Tokenizer + Transformer (FT-Transformer). FT-Transformer first projects numerical and categorical features onto the embedding space respectively, followed by applying a stack of Transformer layers to the embeddings. Another line of work leverages set function learning for time series [Horn et al., 2020, Tipirneni and Reddy, 2022]. The set function representation addresses the prevalent issues in time series data such as missing information and irregular time intervals. Tipirneni and Reddy [2022] treat time series as a set of observation triplets, defined as (time, feature, value). They developed a novel Continuous Value Embedding (CVE) mechanism to embed time and value in the triplet. They also applied self-supervised learning with forecasting as the predictive objective to learn robust feature-level representations.

B Model Parameters and Experiment Setup

Training setup The pipeline of our method is shown in Figure 1a. In the pre-training stage, we use the self-supervised objectives CBOW or MLM to learn feature level embeddings. In the fine-tuning stage, we pool the pre-trained feature level embeddings to get the time-step level embeddings, which are then encoded by the Transformer model for clinical prediction.

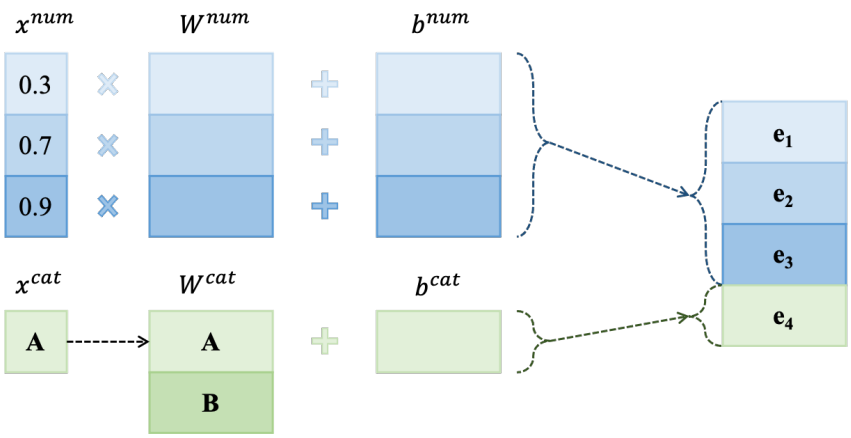


Figure 3: Feature Tokenizer model from Gorishniy et al. [2021]

batch size	LR	feature_dim	depth	heads	AUPRC	AUROC
8	0.001	64	1	1	36.6 ± 0.5	89.7 ± 0.1
8	0.001	64	2	2	35.8 ± 0.1	89.5 ± 0.1
8	0.001	128	1	1	36.1 ± 0.2	89.5 ± 0.1
8	0.001	128	2	2	34.7 ± 0.2	89.3 ± 0.1
8	0.0001	64	1	1	37.9 ± 0.2	90.1 ± 0.1
8	0.0001	64	2	2	37.9 ± 0.2	90.1 ± 0.1
8	0.0001	128	1	1	37.8 ± 0.4	90.1 ± 0.1
8	0.0001	128	2	2	37.7 ± 0.6	90.0 ± 0.3
16	0.001	64	1	1	37.0 ± 0.6	89.9 ± 0.1
16	0.001	64	2	2	36.8 ± 0.3	89.7 ± 0.1
16	0.001	128	1	1	36.5 ± 0.1	89.8 ± 0.1
16	0.001	128	2	2	35.0 ± 1.0	89.5 ± 0.3
16	0.0001	64	1	1	37.8 ± 0.1	90.1 ± 0.1
16	0.0001	64	2	2	37.9 ± 0.2	90.2 ± 0.1
16	0.0001	128	1	1	$\mathbf{38.4} \pm 0.1$	$\mathbf{90.3} \pm 0.0$
16	0.0001	128	2	2	38.0 ± 0.4	90.1 ± 0.1

Table 2: Random search results for fine-tuning parameters. AUPRC and AUROC is on the validation set of MIMIC-III decompensation prediction task. We report mean and standard deviation from three runs.

Module	Parameter	Value
Fine-Tune	batch size	16
	learning rate	0.0001
	feature dimension	128
	depth	1
	num_heads	1
	pooling	max
CBOW	batch size	256
	learning rate	0.01
	feature dimension	256
MLM	batch size	512
	learning rate	0.0001
	feature dimension	128
	depth	2
	num_heads	1

Table 3: Training Hyper-parameters.

Abbr.	Name
DBP	diastolic blood pressure
FIO	fraction of inspired oxygen
HR	heart rate
MBP	mean blood pressure
OS	oxygen saturation
RR	respiratory rate
SBP	systolic blood pressure
Temp	temperature
CRR	capillary refill rate
GCST	Glascow coma scale total
GCSEO	Glascow coma scale eye opening
GCSMR	Glascow coma scale motor response
GCSVR	Glascow coma scale verbal response

Table 4: Clinical feature name abbreviations.

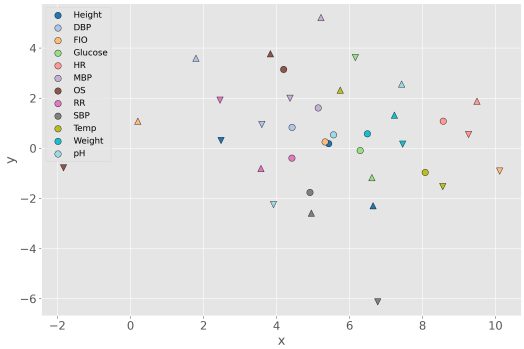


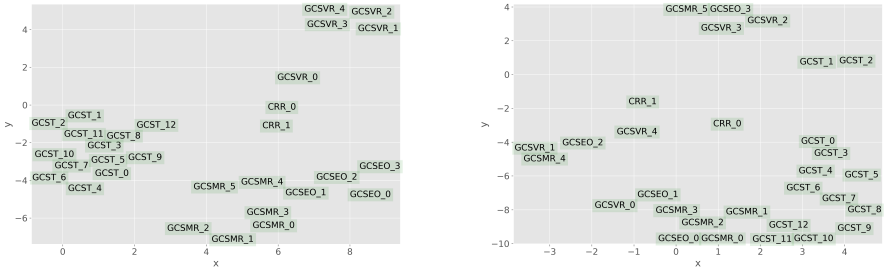
Figure 4: T-SNE visualization, with the perplexity value set to 15, of numerical feature embeddings from FTT.

C Additional Results

Limited Labeled Data Analysis To further explore the effect of pre-trained feature embeddings on the decompensation prediction task, we gradually reduce the number of labeled data from 100% to 1% during fine-tuning. The results are shown in Table 5.

Ablation Study on CBOW Besides the traditional CBOW model, we also explored adding information from the previous time step to predict current feature values. The results are shown in Table 6. It turns out that the adapted CBOW pre-training does not improve the downstream task performance.

Additional Explanations Although we observe good alignment between learnt clinical feature embeddings and prior clinical knowledge, the performance on the decompensation and mortality prediction tasks is unexpectedly not improved when leveraging the pre-trained embeddings. We believe it is necessary to conduct a series of experiments on various downstream tasks to see whether our pre-trained embeddings can help. Besides, the pre-trained clinical embeddings could be combined with higher-level embeddings (e.g. time-series level or patient-level embeddings) to further improve performances on downstream tasks.



(a) **FTT** (b) **CBOW**

(c) **MLM**

Figure 5: T-SNE visualization, with the perplexity value set to 15, of categorical feature embeddings from FTT, CBOW and MLM.

Labels	Models	AUPRC	AUROC
100%	Transformer	34.4 ± 0.4	91.2 ± 0.1
	FTT	36.4 ± 0.2	91.6 ± 0.1
	CBOW	36.3 ± 0.4	91.4 ± 0.1
	MLM	36.2 ± 0.1	91.6 ± 0.1
50%	Transformer	33.1 ± 0.1	90.8 ± 0.1
	FTT	35.8 ± 0.3	91.2 ± 0.1
	CBOW	34.8 ± 0.2	91.1 ± 0.1
	MLM	34.0 ± 0.5	91.1 ± 0.1
10%	Transformer	31.2 ± 0.2	90.0 ± 0.1
	FTT	31.3 ± 0.3	89.8 ± 0.5
	CBOW	28.9 ± 0.6	88.6 ± 0.2
	MLM	31.3 ± 0.3	89.4 ± 0.1
1%	Transformer	22.2 ± 1.0	85.5 ± 1.0
	FTT	8.7 ± 7.0	63.2 ± 18.5
	CBOW	19.1 ± 1.8	83.6 ± 0.4
	MLM	11.2 ± 1.0	78.4 ± 2.5

Table 5: Performance on the decompensation task for different models with decreasing labeled data. Mean and standard deviation are reported over three runs.

Use_previous	AUPRC	AUROC
False	36.3 ± 0.4	91.4 ± 0.1
True	35.8 ± 0.1	91.4 ± 0.1

Table 6: Comparing the performance of adapted CBOW objectives on decompensation task.