HHS Public Access

Author manuscript

ACM BCB. Author manuscript; available in PMC 2022 August 10.

Published in final edited form as:

ACM BCB. 2022 August ; 2022: . doi:10.1145/3535508.3545541.

Supervised Pretraining through Contrastive Categorical Positive Samplings to Improve COVID-19 Mortality Prediction

Tingyi Wanyan,
Population Health Sciences, Weill Cornell Medicine, New York, NY, USA

Mingquan Lin,
Population Health Sciences, Weill Cornell Medicine, New York, NY, USA

Eyal Klang,
Icahn School of Medicine at Mount Sinai, New York, NY, USA

Kartikeya M. Menon,
Icahn School of Medicine at Mount Sinai, New York, NY, USA

Faris F. Gulamali,
Icahn School of Medicine at Mount Sinai, New York, NY, USA

Ariful Azad,
Intelligent Systems Engineering, Indiana University, Bloomington, Bloomington, IN, USA

Yiye Zhang,
Population Health Sciences, Weill Cornell Medicine, New York, NY, USA

Ying Ding,
School of Information, University of Texus Austin, Austin, TX, USA

Zhangyang Wang,
Electrical and Computer Engineering, University of Texus Austin, Austin, TX, USA

Fei Wang,
Population Health Sciences, Weill Cornell Medicine, New York, NY, USA

Benjamin Glicksberg[*],
Icahn School of Medicine at Mount Sinai, New York, NY, USA

Yifan Peng[*]
Population Health Sciences, Weill Cornell Medicine, New York, NY, USA

Abstract

tiw4003@med.cornell.edu .
[*]Both authors contributed equally to the paper.

Clinical EHR data is naturally heterogeneous, where it contains abundant sub-phenotype. Such diversity creates challenges for outcome prediction using a machine learning model since it leads to high intra-class variance. To address this issue, we propose a supervised pre-training model with a unique embedded k-nearest-neighbor positive sampling strategy. We demonstrate the enhanced performance value of this framework theoretically and show that it yields highly competitive experimental results in predicting patient mortality in real-world COVID-19 EHR data with a total of over 7,000 patients admitted to a large, urban health system. Our method achieves a better AUROC prediction score of 0.872, which outperforms the alternative pre-training models and traditional machine learning methods. Additionally, our method performs much better when the training data size is small (345 training instances).

Keywords

Sub-phenotype; Intra-class variance; Pre-training; Self-supervised Learning; Supervised Contrastive Learning; mortality prediction

1 INTRODUCTION

Electronic health records (EHRs) contain large amounts of longitudinal, heterogeneous data generated by clinical activities, ranging from structured data (e.g., disease diagnoses, laboratory test results, and vital signs), to unstructured clinical notes and medical images. It provides great potential for improving human health and clinical care, such as patient health trajectory modeling [8], disease inference [2], and clinical decision support systems [16].

In recent years, many machine learning techniques, including deep learning, have been leveraged to derive insights from EHR data [6, 12, 18, 21, 28]. However, the intra-class heterogeneous nature of EHRs presents one barrier to the training and deployment of state-of-the-art machine learning models. Specifically, EHR data typically possess different sub-phenotype within the same class group. For example, COVID-19 patients may present different sub-phenotype regarding distinctive respiratory parameters [3]; Sepsis patients have various acute kidney injury (AKI) associated phenotypes [4]. Such diversity within the cohort population leads to a high intra-class variance in etiology and presentation.

It is not a trivial task to handle the intra-class variance in the deep learning models. Simple cross-entropy loss based on label information would not capture such diverse data heterogeneity because it could easily produce a poor margin for a decision boundary [19], thus resulting in poor generalization. In recent years, supervised contrastive learning has provided a promising solution because it can generate discriminative features by pulling together positive pairs from the same class and pushing apart the negative pairs from different classes [5]. Nevertheless, it suffers from class collapse, where each example in the same class has the same representation. Therefore, it cannot distinguish the latent subphenotypes within the patient of the same condition.

To overcome this issue, we propose a new Embedding-Based K-NN Positive Sampling Contrastive Learning (EKPS-CL) method to model EHR data intra-class variance (Figure 1). Specifically, we first construct a KNN (K Nearest Neighborhood) graph from the

patients with the same conditions. Then we apply a supervised contrastive learning strategy to pull examples from the same neighbor closer together than examples from different neighbors. Compared to previous studies, our model can learn a "spread out" representation to distinguish subphenotype classes. To optimize the number of neighbors in EKPS-CL , we adopt an Expectation-Maximization (EM) strategy. In the E-step, we perform unsupervised learning to construct a KNN. In the M-step, we sample "positive" and noisy "negative" samples out of the KNN graph for contrastive learning. To extrinsically evaluate our method, we apply EKPS-CL to an important problem: predicting the mortality of patients with COVID-19. Experiments on the newly collected data from Mount Sinai Hospital show that our pre-training model can largely increase the prediction accuracy (~ 7% on average). In addition, it outperforms the state-of-the-art pre-training methods (~ 2% increment).

Our contributions can be summarized in the following three-fold. (1) We propose a new pre-training loss function that could lead to a better embedding representation by approximating EHR data intra-class variance. (2) Following the designed loss function, we propose a pre-training algorithm called EKPS-CL that efficiently learns the embedding representation. (3) We empirically evaluate our pre-train model on a use case of predicting the mortality of patients with COVID-19 and demonstrate its superior performance against both Simple contrastive learning (SimCLR) and supervised contrastive learning (SupCLR), as well as other traditional machine learning baselines.

2 RELATED WORK

Phenotype intra-class heterogeneity and sub-phenotypes.

Clinical phenotypes are often expressed heterogeneously, meaning the same disease can have different lab test values or diagnostic codes. For example, individuals with severe COVID-19 have large intra-class variance in terms of etiology and presentation [23]. Patterns in this phenomenon can be considered sub-phenotypes of a disease. Exploring different sub-phenotypes is valuable to precision medicine and can enhance the performance of the predictive tasks, leading to more personalized recommendations. There is a large body of work exploring computational methods for subphenotyping, such as Parkinson's disease [17], scleroderma [20], and Glioblastoma [25]. Methods such as multi-task learning and hierarchical models [1, 24] have been employed to better capture subtype patterns. This discovery suggests the importance of capturing EHR data heterogeneity in machine learning systems.

Self-supervised contrastive learning (SimpCLR) and Supervised contrastive learning (SupCLR).

Self-supervised contrastive learning characterizes intra-class clusters by employing contrastive learning within classes. In this method, clusters of the same-class points have a single positive sample as an augmented anchor. This anchor takes advantage of the intrinsic distribution without relying on pre-labeled class information [14].

Contrastive learning can also be fully-supervised [14]. The idea is to pull "similar points" (or points belonging to the same class) together, while simultaneously pushing apart

"dissimilar points" (or points belonging to different classes) in the embedding space. Studies show that SupCLR performs consistently better than SimpCLR and cross-entropy loss on large-scale classification problems [14, 26]. It also yields superior robustness to noise and unseen corruptions during testing [5]. For example, Khosla et al. [14] proposed a unified loss function that can be viewed as the generalization of both triplet [27] and *N*-pair [22] losses. Their loss is less sensitive to hyperparameters, which can provide consistent boosts for accuracy for different datasets, and is robust to natural corruptions.

Nevertheless, previous literature either performs data augmentation to create a similar anchor and positive examples (in the unsupervised setting) [7, 11], or randomly samples examples from the same class (in the supervised setting) [13, 14]. They suffer from class collapse, where each example in the same class has the same representation. Therefore, it cannot distinguish the latent subphenotypes within the patient of the same condition. Our technical novelty in this work is to automatically consider the underlying feature clusters when sampling data points for contrastive learning. As a result, our model can learn a "spread out" representation to distinguish subphenotype classes.

3 MATERIALS AND METHODS

3.1 Task definition

While our method can be used on any task, in this study, we focus on the study of dynamic prediction in the medical field. Nowadays, there is great interest in prognostic models and their application to personalized medicine. In the dynamic prediction, the survival probabilities are dynamically updated as additional longitudinal information is recorded.

More formally, for a new subject i, we have available measurements up to time point t. We are interested in $Pr(T_i^* \geq u \mid T_i^* > t)$, where $u \geq t$ and T_i^* is the true event time. In the discrete context, we divide the continuous time into joint intervals $V = (t_{l-1}, t_l]$ where t_0 and t_T are the first and last observations interval boundaries (Figure 2). Our goal is to predict the mortality probability at time t_u with longitudinal features in the observation window $(t_0, t_u]$.

3.2 Overall architecture

Our overall architecture is presented in Figure 1. We first perform the proposed pre-training model which learns the initial parameters for the longitudinal model. We then fine-tune it in the downstream task.

Because of the sequential nature of the problem, we use the Long Short-Term Memory (LSTM) model to capture the time-variant effect of each feature over time. Specifically, we apply a two layers LSTM structure, followed by a dropout layer for both the pretraining and downstream tasks. The input is given by the feature x_i of patient i (such as lab tests and vital signals). The output is the hidden state output for the last time step (t_u). We then concatenated it with the static features (such as demographics) to form the final embedding output for the downstream application (Figure 3).

3.3 Supervised pre-training through contrastive categorical positive samplings

In this section, we discuss the proposed pre-training model by first introducing the loss function, followed by describing the algorithm.

Based on the heterogeneity of EHR data, we assume that there exist sub-phenotype category groups within the same class group. we denote this set of category groups as as $C = \{c_1, \ldots c_k\}$. Let $h_i = f(x_i)$ be the output of the embedding function f(.) with the input feature x_i. In this study, h_i is the final embedding output in Figure 3.

3.3.1 Objective Function.—To make the pre-train embedding distribution approximate a multi-subcategory distribution, we design our system to maximize the conditional joint probability of a multi-categorical model in Equation (1), in which θ is the model parameters, h_i is the embedding vector representation of a data instance, C is a set of categories, c_j is an indicator that h_i belongs to subcategory j, and n is the batch size.

$$\begin{aligned}\theta^* &= \arg\max_{\theta} \sum_{i=1}^{n} \sum_{j=1}^{k} c_j \log p(h_i \mid c_j \in C; \theta) \\ &= \arg\max_{\theta} \sum_{i=1}^{n} \sum_{j=1}^{k} c_j \log \frac{p(h_i, c_j; \theta)}{\sum_{h \in H} p(h, c_j)}\end{aligned} \tag{1}$$

Explicitly computing the denominator term would be intractable since it requires integrating over the whole embedding space H. Therefore, we resort to using Noise Contrastive Estimation (NCE) to approximate the objective function and optimize θ [10]. Since the objective function relates to multi-categorical distribution, which is slightly different from the original NCE loss, we derive our NCE optimization as follows.

Given a set of n samples with one "positive" sample x_i from category c_j, and $n-1$ noise samples from the other categories $C/\{c_j\}$. The categorical conditional probability is:

$$p(c_j \mid h_i, \theta) = \frac{p(h_i \mid c_j, \theta)}{p(h_i \mid c_j, \theta) + \frac{1}{\mid C \mid -1} \sum_{c_m \in C / \{c_j\}} p(h_i \mid c_m, \theta)}$$

Thus, the objective function of the joint conditional categorical probability is:

$$L(\theta) = \sum_{i=1}^{n} \sum_{j=1}^{k} c_j \log p(c_j \mid h_i, \theta) \tag{2}$$

Semantically, optimizing $L(\theta)$ is to maximize a probability density ratio between the conditional probability where the data embedding h_i is generated from the actual category c_j, against the data embedding generated from the noise categories $C/\{c_j\}$, thus lead to an approximation of maximizing (1).

To model the categorical conditional probability in (2), we use the log-bilinear model $\exp(h_i \cdot \bar{c}_j) \propto p(h_i/c_j, \theta)$, where $\bar{c}_j$ is a context embedding vector of c_j, and will be discussed later. The final form of our objective function is:

$$L = \sum_{i=1}^{n} \sum_{j=1}^{k} c_j \log \frac{\exp(h_i \bar{c}_j)}{\exp(h_i \bar{c}_j) + \frac{1}{|C|-1} \sum_{c_m \in C / \{c_j\}} \exp(h_i \bar{c}_m)} \tag{3}$$

3.3.2 Learning algorithm.—There are many ways to model $\bar{c}$. In this study, we find that a simple sampling could achieve good performance. To optimize (3), we adopt an Expectation-Maximization (EM) strategy. In the E-step, we perform unsupervised learning to construct a KNN (*K*-nearest Neighborhood) graph out of the whole data embeddings (Figure 4). The similarity criteria between the embeddings are based on their inner product. In the M-step, we sample "positive" and noisy "negative" samples out of the KNN graph and optimize (3). Our algorithm is presented in Algorithm 1.

Algorithm 1 Embedding-based K-nearest Neighborhood Sampling Contrastive Learning

Input: Longitutinal EHR features
Output: Pretrained systemm with parameters tuned
1: Initialize system parameters θ
2: **for** each epoch **do**
3: For each class group (label 1 or 0), Compute similarities between all pairs of embedding feature representations based on their inner product, and build KNN graph from it .
4: **while** not converged **do**
5: Sample a mini-batch training patients $P \in P_{all}$
6: **for** each $p \in P$ **do**
7: Sample 1 ''positive" sample data $p_k^+ \in P_{all}$ that have the same label as p, and are connected to node p in the KNN graph.
8: **end for**
9: Optimize L in equation 3
10: **end while**
11: **end for**
12: **return** Pre-trained deep learning system

3.4 Downstream dynamic prediction

The objective is a binary cross-entropy loss for predicting survival probability:

$$L_b = -y \log(\hat{y}) + (1-y) \log(1-\hat{y}) \tag{4}$$

where y is the actual label and $\hat{y}$ is the predicted probability from the projection layer. As shown in Figure 3, the cross entropy loss is calculated based on the output of the Dense Projection Layer.

4 EXPERIMENTS

While our method can be used on any dynamic prediction task, in this study, we focus on the problem of mortality prediction of patients with COVID-19. Early prediction of COVID-19 mortality is important since it can alleviate the burden on healthcare systems and help identify efficient early detection of patient deterioration which is important for allocating limited resources [9].

4.1 Dataset

We obtained the EHR data of COVID-19 patients from five hospitals within the Mount Sinai Health System located in New York City. The EHR data collected contains the following patient data: COVID-19 status, Intensive Care Unit (ICU) status, demographics, lab test results, vital signs, comorbid diseases, and outcome (e.g., mortality, discharge). Lab tests and vital signs were measured at multiple time points along the hospital course. We included nine frequently measured vital signs: heart rate, respiration rate, pulse oximetry, blood pressure (diastolic and systolic), temperature, oxygen saturation, height, and weight. We also selected 76 lab tests that were both commonly measured and relevant to COVID-19. For the static features, we included age, gender, and race as demographics and 12 comorbid diseases: atrial fibrillation, asthma, coronary artery disease, cancer, chronic kidney disease, chronic obstructive pulmonary disease, diabetes mellitus, health failure, hypertension, stroke, alcoholism, and liver disease. Comorbid diseases were considered from their presence at admission to the hospital and defined via ICD9/10-CM codes collapsed by Phecode. This study has been approved by the Institutional Review Board at the Icahn School of Medicine at Mount Sinai (IRB- 20-03271).

Overall, we obtained 7,067 patients who tested positive for COVID-19 and were hospitalized (~23% mortality rate). Table 1 lists the patient statistics for patient commodities and vital signals. Details of the lab tests can be found in the Appendix.

4.2 Baseline methods

For baseline comparisons, we divide the methods into two groups. The first group of methods does not apply pre-training. Here, we listed 4 traditional machine learning models that are commonly used on prediction tasks: Logistic Regression [18], Random Forest [12], Support Vector Machine (SVM) [29], and XGboost [28]. The input features for these baseline models are the averaged feature values (labs and vitals) in the observation window. Additionally, we included an LSTM model with Cross-Entropy Loss without pre-training.

The second group contains two pre-training methodologies: Sim-CLR [7] and SupCLR [14]. SimCLR requires the augmented data generated from the training instance. Therefore, we generated the positive sample by picking a temporal embedding from our longitudinal structure at a random time step. For SupCLR, we implemented the same procedure from the original paper. Note that SupCLR is different from our methods in that it samples “positive” instances from the same binary classification group (label 1 or 0) rather than from the sub-phenotype category groups within the same group.

4.3 Experimental settings

We split our data into 70% for training, 10% for validation, and 20% for testing. The longitudinal data (i.e., features with multiple values), specifically lab tests and vital signs, were binned and averaged within 4-hour windows across their hospitalization. We pre-process the longitudinal and static features by considering the values between 0.5 and 99.5 percentile to remove any inaccurate measurement, we then normalize the data by calculating the standard score (Z score). For categorical comorbid data, we use one-hot encoding representation. Numerical data with missing values are imputed with zeros.

All analyses were performed using TensorFlow 1.15.1 and utilized the Adam optimizer [15]. We set the batch size to be 256, with 30 training epochs. The embedding dimension is set to be 100. We used two NVIDIA 2080 TI in our experiments.

For the evaluation metric, we reported the area under the receiver-operating characteristic (AUROC) and the area under the precision, recall curve (AUPRC). We used 10 bootstrap samples to obtain a distribution of the evaluation metrics and reported 95% confidence intervals.

4.4 Results and Discussion

4.4.1 Prediction performance.—Table 2 shows the performance to predict mortality of patients with COVID-19 using the HER within 8, 12, 24, and 48 hours after hospital admission. First, we observed that the prediction performance (AUC score) of the baseline models without pre-training is similar (around 0.7), with the random forest model showing slightly better performance. In contrast, our pre-training model largely increases the performance (above 7% increment) against the models without pre-training. Secondly, among the pre-training methodologies, EKPS-CL achieves the best performance through all time periods (around 3% over SimCLR and 2% over SupCLR), which verifies our hypothesis and analysis in Section 3.3.

4.4.2 Effect of the learning behaviors.—Next, we compared different learning behaviors (Figure 5), where we recorded the AUC scores on the validation set for every epoch. In this experiment, we compared SEKPS-CL to SimCLR and SupCLR, as well as the model with a single cross-entropy (CE) loss but not pre-training. The learning curve shows that, compared to the model with only a CE loss, models with pre-training consistently achieved higher accuracy at every epoch. More importantly, SEKPS-CL achieved a higher AUC score along with each epoch than SimCLR and SupCLR, showing the effects of categorical positive sampling.

4.4.3 Effect of the training size.—We performed an additional study to evaluate the robustness of different models (Figure 7). In this evaluation scenario, we varied the training data size but kept the testing and evaluation data set unchanged. The purpose is to assess how our pre-training model would help on remaining stable performance against baselines under situations when training data sets are limited. Specifically, we compared performance at $N = [345, 945, 1945, 2945, 3450]$ where N is the training data size. We used random down-sampling to make the data unbiased.

4.4.4 Effect of k in KNN.—We also evaluated the model performance on different k in KNN (Figure 6). The model reaches peak performance when $k = 5$.

Figure 7 shows the AUC results corresponding to different training sample sizes. The performance of models without pre-training substantially decreases as the training size becomes smaller (Totally 12% decrement). On the other hand, models with pre-training have much more stable performance, where the performance decrement for SimcCLR, SupCLR, and SEKPS-CL are 0.23, 0.21, and 0.20, respectively. In addition, SEKPS-CL pre-training consistently outperforms SimCLR and SupCLR within each training set (different in size). This observation suggests the effectiveness of our pre-training algorithm.

4.4.5 Discussion.—From our experiments, we observed that models that do not apply pre-training, such as models that directly adopt CE loss, tend to make the embeddings separate into two distributions where each represents a class group. When adopting supervised contrastive pre-training, the two-class distribution centers are pushed further away than using the use CE loss only. In the meantime, SimCLR is a self-supervised pre-training strategy, which maximizes the data intra-class variance. Thus, when adopting this strategy, the effect is similar to generating embeddings that approximate multi sub-phenotype categories.

Our pre-training method (EKPS-CL) combines the advantages of both SimCLR and SupCLR. First, we performed the supervised sampling (sample "positive" samples from the same class group). This brings the strength of SupCLR by pushing away the two-class distribution centers. Secondly, though the "positive" samples are picked from the same class group, we selectively sample these positive samples from the sub-categories by constructing a KNN graph. Hence, this procedure is similar to recognizing the intra-class heterogeneous which is consistent with the effect of SimCLR. As evidenced by our experiments, combining the two approaches significantly improves the downstream prediction performance.

5 CONCLUSION

In this work, we propose a pre-training algorithm designed based on the unique EHR data heterogeneous characteristic. We evaluated the algorithm for predicting the mortality of COVID-19 patients and demonstrated the superior performance of our algorithm over alternative baselines. Our analysis and results showed the great potential to improve prediction performance by designing pre-training models that consider both the class label information as well as the EHR data heterogeneity.

ACKNOWLEDGMENT

This work is supported by the National Library of Medicine under Award No. 4R00LM013001.

APPENDIX

Table 3:

Statistics of the laboratory test

Laboratory Values (median, IQR)	Overall		Mortality	
Albumin (g/dL)	2.5	(0.8)	2.1	(0.75)
Alkaline Phosphatase (units/L)	74.0	(40.0)	80.5	(41.5)
Alanine Transaminase (units/L)	27.0	(31.0)	28.0	(27.0)
Anion Gap (mEq/L)	10.0	(3.25)	11.68	(4.22)
Partial Thromboplastin Time (s)	32.7	(9.09)	35.3	(13.56)
Aspartate Aminotransferase (units/L)	32.0	(26.0)	39.0	(34.0)
Atypical lymphocyte percentage (Band count	3.0	(3.0)	4.0	(5.0)
Direct Bilirubin (mg/dL)	0.3	(0.2)	0.3	(0.3)
Total Bilirubin (mg/dL)	0.5	(0.4)	0.5	(0.45)
B-Type Natriuretic Peptide (ng/mL)	148.02	(398.16)	87.6	(237.39)
Blood Urea Nitrogen (mg/dL)	17.5	(22.0)	36.0	(37.5)
C Reactive Protein (mg/L)	52.61	(84.68)	104.6	(128.5)
Ionized calcium (mg/dL)	1.15	(0.09)	1.14	(0.12)
Calcium (mg/dL)	8.0	(0.75)	7.65	(0.9)
Chloride (mEq/L)	102.0	(6.0)	103.0	(9.0)
Creatine Phosphokinase (units/L)	103.0	(249.25)	142.0	(321.75)
Creatine Kinase-MB (units/L)	3.35	(4.38)	3.2	(3.1)
Bicarbonate (mEq/L)	22.6	(5.0)	20.65	(5.65)
Creatinine (mg/dL)	0.89	(0.79)	1.4	(2.02)
D-Dimer (ng/mL)	1.42	(1.88)	2.31	(2.33)
Ferritin (ng/mL)	687.0	(224.5)	1011.0	(1555.5)
Fibrinogen (mg/dL)	520.0	(240.5)	526.0	(242.75)
Glucose (mg/dL)	100.0	(45.0)	124.0	(61.0)
Hematocrit (%)	34.2	(10.7)	31.9	(11.72)
Hemoglobin (g/dL)	11.0	(3.65)	10.2	(3.9)
International Normalised Ratio	1.15	(0.2)	1.2	(0.3)
Interleukin-6 (pg/mL)	46.4	(136.88)	143.7	(278.15)
Iron (mcg/dL)	25.0	(21.0)	33.0	(29.75)
Lactate (mmol/L)	1.4	(0.65)	1.58	(0.85)
Lactate Dehydrogenase (U/L)	371.0	(193.12)	517.0	(248.0)
Lymphocyte Percentage (%)	10.1	(10.9)	4.95	(4.86)
Lymphocyte Count	0.9	(0.6)	0.6	(0.5)
Mean Corpuscular Hemoglobin Concentration (g/dL)	29.6	(2.85)	29.8	(2.85)
Mean Corpuscular Volume (fL)	90.4	(8.0)	91.3	(8.75)
Mean Platelet Volume (fL)	8.2	(1.5)	8.6	(1.64)
Monocyte Percentage (Monocyte Count	0.4	(0.3)	0.4	(0.35)
Neutrophil Percentage (Neutrophil Count	5.0	(4.05)	7.6	(5.3)
Partial pressure of carbon dioxide (mmHg)	39.5	(8.5)	39.0	(11.62)
pH	7.37	(0.08)	7.32	(0.14)

Laboratory Values (median, IQR)	Overall		Mortality	
Platelets	211.0	(133.0)	178.0	(109.5)
Partial pressure of oxygen (mmHg)	40.0	(16.0)	39.0	(14.5)
Potassium (mEq/L)	4.0	(0.6)	4.15	(0.85)
Prothrombin time (s)	14.4	(2.0)	15.1	(2.22)
Serum protein (g/dL)	6.0	(1.05)	5.6	(1.1)
Red Blood Cell Count	3.78	(1.2)	3.48	(1.3)
Red Blood Cell Distribution Width (Sodium (mEq/L)	137.5	(5.0)	139.0	(7.0)
Total iron binding capacity (mcg/dL)	163.0	(76.25)	203.0	(100.0)
Transferrin Saturation (Troponin I (ng/mL)	0.06	(0.15)	0.08	(0.25)
White Blood Cells (uL)	7.1	(4.5)	9.7	(6.35)

REFERENCES

[1]. Alaa AM, Yoon J, Hu S, and van der Schaar M. 2018. Personalized Risk Scoring for Critical Care Prognosis Using Mixtures of Gaussian Processes. IEEE Transactions on Biomedical Engineering 65, 1 (2018), 207–218. 10.1109/TBME.2017.2698602 [PubMed: 28463183]

[2]. Austin Peter C, Tu Jack V, Ho Jennifer E, Levy Daniel, and Lee Douglas S. 2013. Using methods from the data-mining and machine-learning literature for disease classification and prediction: a case study examining classification of heart failure subtypes. Journal of clinical epidemiology 66, 4 (2013), 398–407. [PubMed: 23384592]

[3]. Bos Lieuwe DJ, Sjoding Michael, Sinha Pratik, Bhavani Sivasubramanium V, Lyons Patrick G, Bewley Alice F, Botta Michela, Tsonas Anissa M, Neto Ary Serpa, Schultz Marcus J, et al. 2021. Longitudinal respiratory subphenotypes in patients with COVID-19-related acute respiratory distress syndrome: results from three observational cohorts. The Lancet Respiratory Medicine 9, 12 (2021), 1377–1386. [PubMed: 34653374]

[4]. Chaudhary Kumardeep, Vaid Akhil, Duffy Áine, Paranjpe Ishan, Jaladanki Suraj, Paranjpe Manish, Johnson Kipp, Gokhale Avantee, Pattharanitima Pattharawin, Chauhan Kinsuk, et al. 2020. Utilization of deep learning for subphenotype identification in sepsis-associated acute kidney injury. Clinical Journal of the American Society of Nephrology 15, 11 (2020), 1557–1565. [PubMed: 33033164]

[5]. Chen Mayee F, Fu Daniel Y, Narayan Avanika, Zhang Michael, Song Zhao, Fatahalian Kayvon, and Ré Christopher. 2022. Perfectly Balanced: Improving Transfer and Robustness of Supervised Contrastive Learning. arXiv preprint arXiv:2204.07596 (2022).

[6]. Chen Peihua and Pan Chuandi. 2018. Diabetes classification model based on boosting algorithms. BMC bioinformatics 19, 1 (2018), 1–9. [PubMed: 29291722]

[7]. Chen Ting, Kornblith Simon, Norouzi Mohammad, and Hinton Geoffrey. 2020. A simple framework for contrastive learning of visual representations. In International conference on machine learning. PMLR, 1597–1607.

[8]. Ebadollahi Shahram, Sun Jimeng, Gotz David, Hu Jianying, Sow Daby, and Neti Chalapathy. 2010. Predicting patient's trajectory of physiological data using temporal trends in similar patients: a system for near-term prognostics. In AMIA annual symposium proceedings, Vol. 2010. American Medical Informatics Association, 192.

[9]. Garrafa Emirena, Vezzoli Marika, Ravanelli Marco, Farina Davide, Borghesi Andrea, Calza Stefano, and Maroldi Roberto. 2021. Early prediction of in-hospital death of COVID-19 patients: a machine-learning model based on age, blood analyses, and chest x-ray score. Elife 10 (2021), e70640. [PubMed: 34661530]

[10]. Gutmann Michael and Hyvärinen Aapo. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Proceedings of the Thirteenth International

Conference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings, 297–304.

[11]. He Kaiming, Fan Haoqi, Wu Yuxin, Xie Saining, and Girshick Ross. 2020. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 9729–9738.

[12]. Hu Chen and Steingrimsson Jon Arni. 2018. Personalized risk prediction in clinical oncology research: applications and practical issues using survival trees and random forests. Journal of biopharmaceutical statistics 28, 2 (2018), 333–349. [PubMed: 29048993]

[13]. Kang Bingyi, Li Yu, Xie Sa, Yuan Zehuan, and Feng Jiashi. 2021. Exploring Balanced Feature Spaces for Representation Learning. In International Conference on Learning Representations.

[14]. Khosla Prannay, Teterwak Piotr, Wang Chen, Sarna Aaron, Tian Yonglong, Isola Phillip, Maschinot Aaron, Liu Ce, and Krishnan Dilip. 2020. Supervised contrastive learning. Advances in Neural Information Processing Systems 33 (2020), 18661–18673.

[15]. Kingma Diederik P and Ba Jimmy. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).

[16]. Kuperman Gilad J, Bobb Anne, Payne Thomas H, Avery Anthony J, Gandhi Tejal K, Burns Gerard, Classen David C, and Bates David W. 2007. Medication-related clinical decision support in computerized provider order entry systems: a review. Journal of the American Medical Informatics Association 14, 1 (2007), 29–40. [PubMed: 17068355]

[17]. Lewis SJG, Foltynie T, Blackwell AD, Robbins TW, Owen AM, and Barker RA. 2005. Heterogeneity of Parkinson's disease in the early clinical stages using a data driven approach. Journal of Neurology, Neurosurgery & Psychiatry 76, 3 (2005), 343–348. [PubMed: 15716523]

[18]. Lorenzoni Giulia, Sabato Stefano Santo, Lanera Corrado, Bottigliengo Daniele, Minto Clara, Ocagli Honoria, De Paolis Paola, Gregori Dario, Iliceto Sabino, and Pisanò Franco. 2019. Comparison of machine learning techniques for prediction of hospitalization in heart failure patients. Journal of clinical medicine 8, 9 (2019), 1298.

[19]. Nar Kamil, Ocal Orhan, Sastry S Shankar, and Ramchandran Kannan. 2018. Crossentropy loss leads to poor margins. (2018).

[20]. Schulam Peter, Wigley Fredrick, and Saria Suchi. 2015. Clustering longitudinal clinical marker trajectories from electronic health data: Applications to phenotyping and endotype discovery. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 29.

[21]. Shickel Benjamin, Tighe Patrick James, Bihorac Azra, and Rashidi Parisa. 2017. Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. IEEE journal of biomedical and health informatics 22, 5 (2017), 1589–1604. [PubMed: 29989977]

[22]. Sohn Kihyuk. 2016. Improved deep metric learning with multi-class n-pair loss objective. In Proceedings of the 30th International Conference on Neural Information Processing Systems. 1857–1865.

[23]. Su Chang, Zhang Yongkang, Flory James H, Weiner Mark G, Kaushal Rainu, Schenck Edward J, and Wang Fei. 2021. Novel clinical subphenotypes in COVID-19: derivation, validation, prediction, temporal patterns, and interaction with social determinants of health. medRxiv (2021).

[24]. Suresh Harini, Gong Jen J, and Guttag John V. 2018. Learning tasks for multitask learning: Heterogenous patient populations in the icu. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 802–810.

[25]. Verhaak Roel GW, Hoadley Katherine A, Purdom Elizabeth, Wang Victoria, Qi Yuan, Wilkerson Matthew D, Miller C Ryan, Ding Li, Golub Todd, Mesirov Jill P, et al. 2010. Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1. Cancer cell 17, 1 (2010), 98–110. [PubMed: 20129251]

[26]. Wanyan Tingyi, Honarvar Hossein, Jaladanki Suraj K, Zang Chengxi, Naik Nidhi, Somani Sulaiman, De Freitas Jessica K, Paranjpe Ishan, Vaid Akhil, Miotto Riccardo, et al. 2021. Contrastive Learning Improves Critical Event Prediction in COVID-19 Patients. arXiv preprint arXiv:2101.04013 (2021).

[27]. Weinberger Kilian Q and Saul Lawrence K. 2009. Distance metric learning for large margin nearest neighbor classification. Journal of machine learning research 10, 2 (2009).

[28]. Yu Bin, Qiu Wenying, Chen Cheng, Ma Anjun, Jiang Jing, Zhou Hongyan, and Ma Qin. 2020. SubMito-XGBoost: predicting protein submitochondrial localization by fusing multiple feature information and eXtreme gradient boosting. Bioinformatics 36, 4 (2020), 1074–1081. [PubMed: 31603468]

[29]. Zhang Xudong, Xiao Jiehao, and Gu Feng. 2019. Applying support vector machine to electronic health records for cancer classification. In 2019 Spring Simulation Conference (SpringSim). IEEE, 1–9.

CCS CONCEPTS

• Theory of computation → Models of learning.

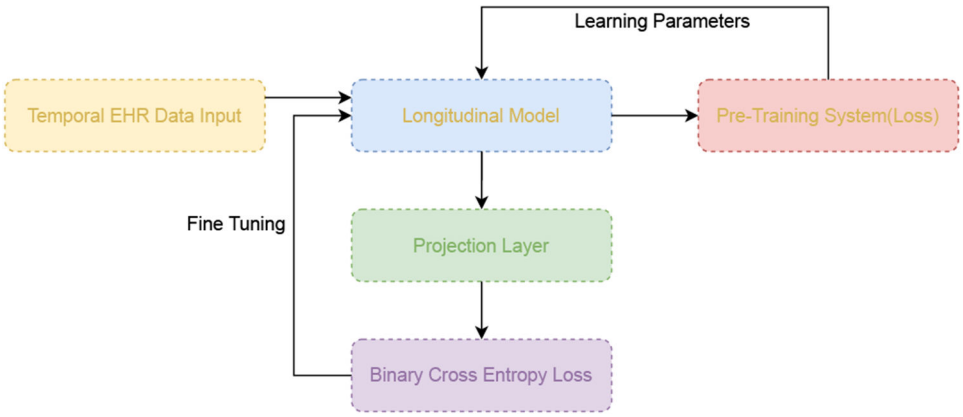

Figure 1:
The overview of the pipeline.

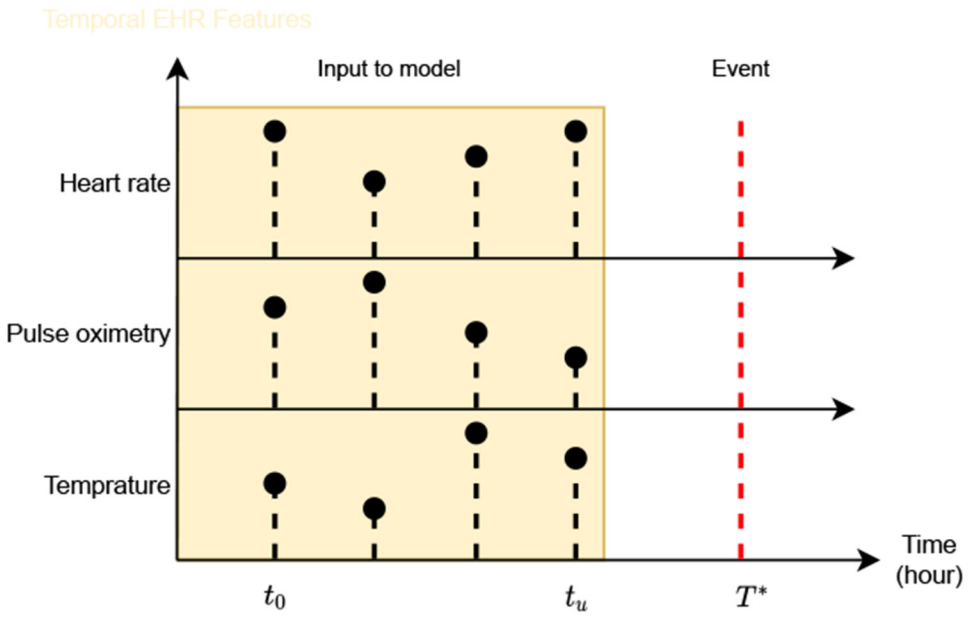

Figure 2:
Example of temporal input.

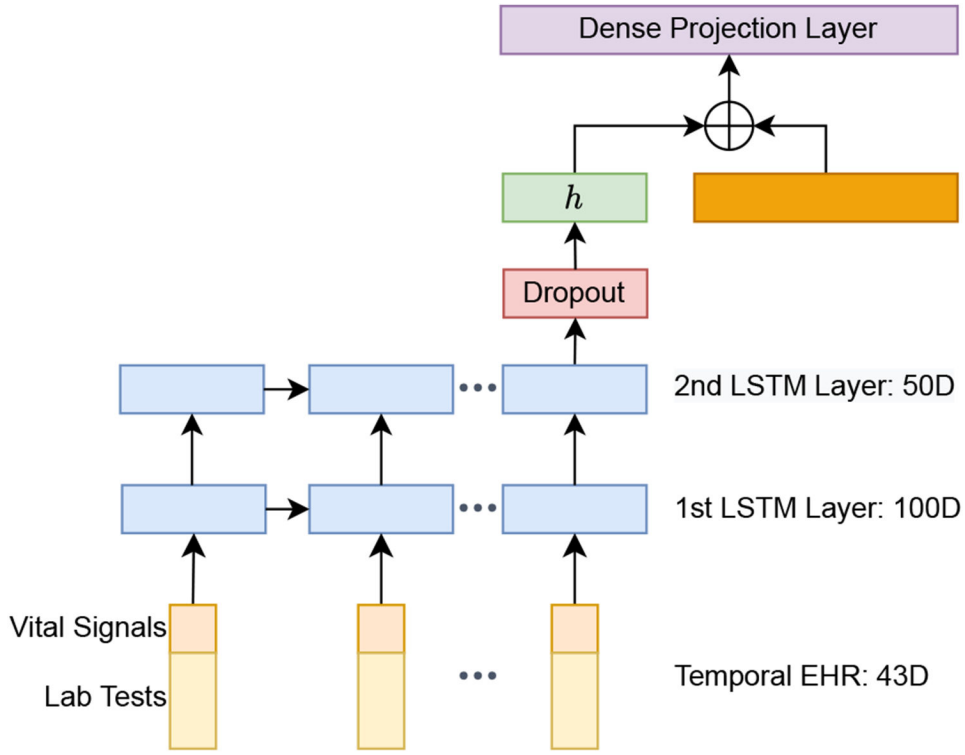

Figure 3:
The longitudinal model structure.

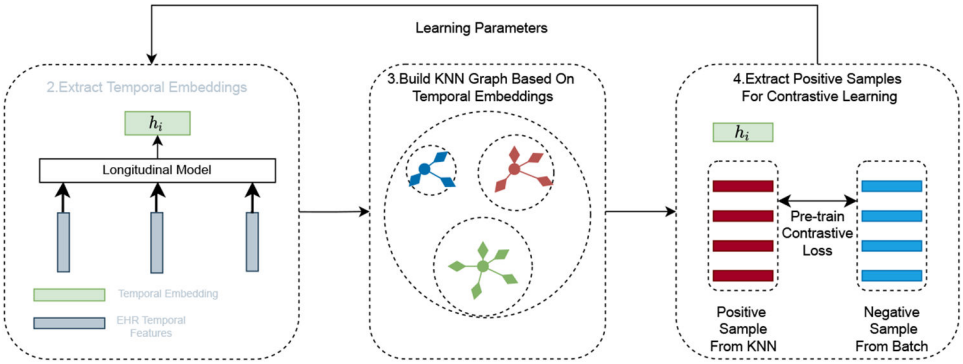

Figure 4:
Pre-training model architecture

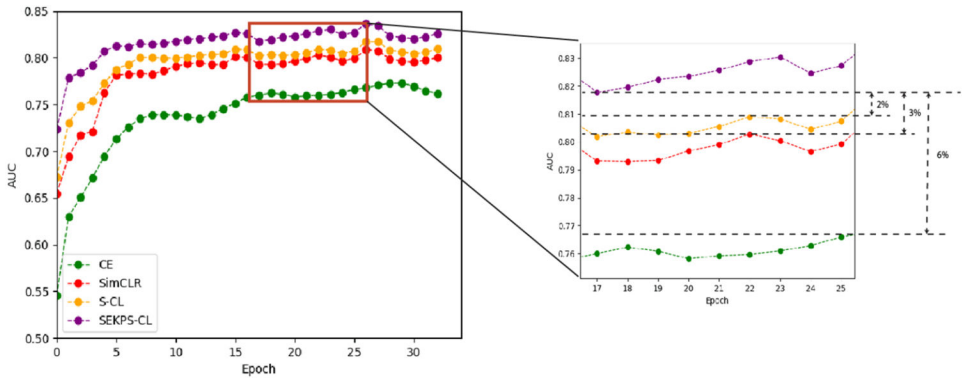

Figure 5:
The learning curves for mortality prediction (observation window=24h).

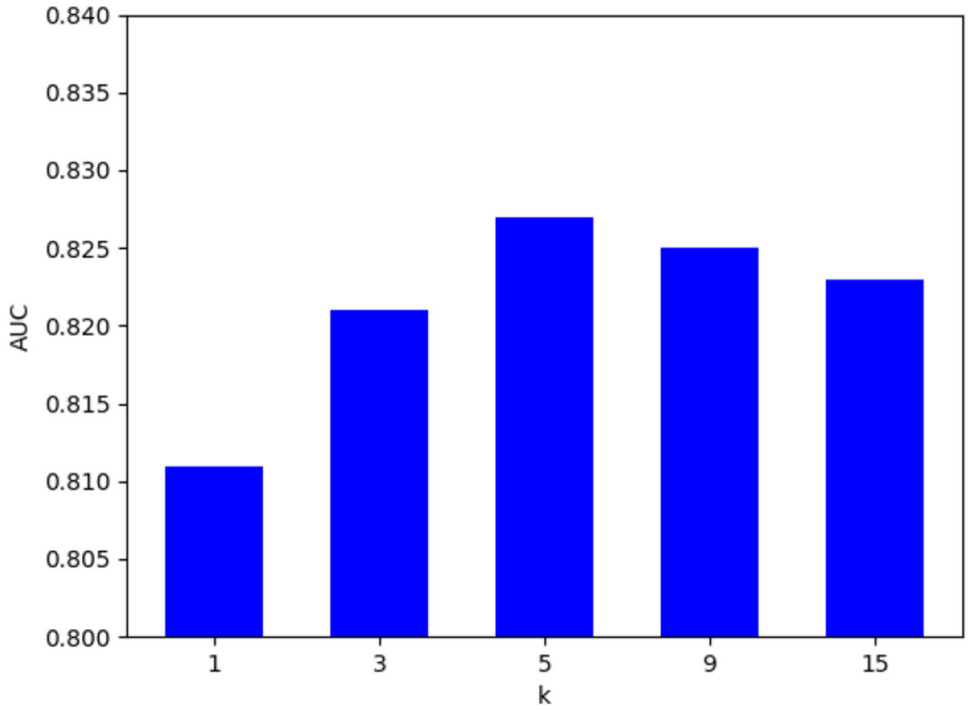

Figure 6:
Predictive Performance on Different *k*

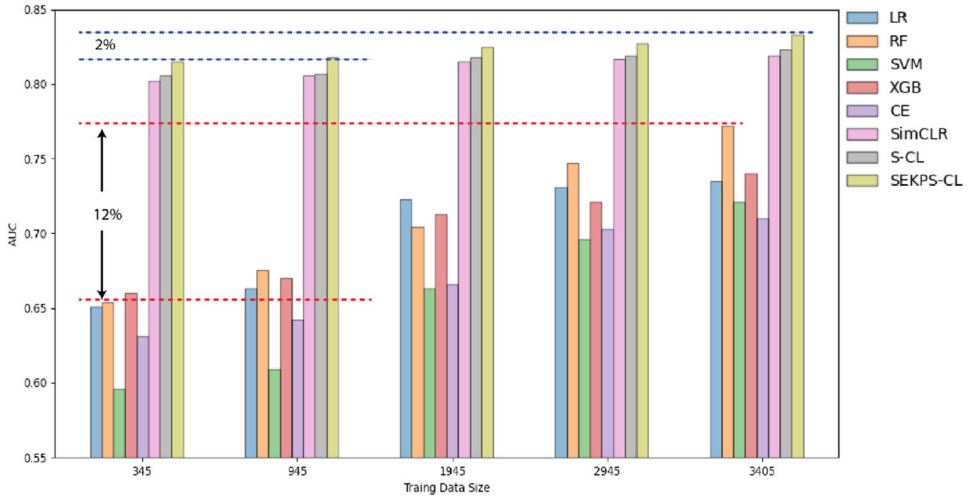

Figure 7:
Prediction performance on mortality prediction with different training data sizes (observation window=24h).

Table 1:

Statistics of the cohort.

	Overall		Mortality	
N	7,067		1,625	
Gender, n (%)				
Male	3,887	(55.3)	924	(58.0)
Female	3,180	(44.6)	701	(42.0)
Age, median (IQR)	64	(24)	74	(19)
Race, n (%)				
Asian	262		59	(4.4)
Black/African American	1,564		357	(26.5)
Other	2,227		501	(37.2)
White	1,443		364	(27.1)
Unknown	216		64	(4.8)
Vital Signs (median, IQR)				
Systolic Blood Pressure, mmHg	114.0	(19.0)	108.0	(20.5)
Diastolic Blood Pressure, mmHg	64.5	(12.0)	59.0	(12.0)
Temperature, °F	97.65	(0.8)	97.6	(1.0)
Pulse Oximetry, %	94.0	(3.0)	93.0	(4.0)
Oxygen Saturation, %	71.47	(28.24)	69.28	(27.05)
Respiration Rate, breaths/min	18.0	(1.5)	19.0	(3.0)
Heart Rate, beats/min	78.0	(17.0)	84.5	(20.0)
Height, (inches)	64.96	(5.99)	64.02	(6.38)
Weight (pounds)	74.4	(27.71)	76.57	(27.13)

Table 2:

The performance of different approaches in predicting risk of mortality for patients with COVID-19.

Model	Observ. window	AUROC (95% CI)	AUPRC (95% CI)	Model	Observ. window	AUROC (95% CI)	AUPRC (95% CI)
LR	8	0.699 (0.687, 0.713)	0.475 (0.447, 0.504)	LR	24	0.732 (0.721, 0.744)	0.524 (0.497, 0.552)
RF	8	0.704 (0.693, 0.715)	0.480 (0.453, 0.501)	RF	24	0.778 (0.768, 0.787)	0.593 (0.573, 0.610)
SVM	8	0.700 (0.692, 0.708)	0.523 (0.511, 0.536)	SVM	24	0.760 (0.743, 0.778)	0.560 (0.541, 0.583)
XG-boost	8	0.675 (0.658, 0.690)	0.460 (0.442, 0.478)	XG-boost	24	0.758 (0.746, 0.769)	0.563 (0.540, 0.587)
LSTM	8	0.685 (0.670, 0.703)	0.489 (0.465, 0.515)	LSTM	24	0.766 (0.751, 0.771)	0.524 (0.503, 0.549)
SimCLR	8	0.754 (0.743, 0.766)	0.554 (0.529, 0.581)	SimCLR	24	0.806 (0.796, 0.814)	0.620 (0.599, 0.643)
SupCLR	8	0.758 (0.740, 0.768)	0.571 (0.553, 0.589)	SupCLR	24	0.809 (0.799, 0.818)	0.627 (0.610, 0.646)
EKPS-CL	8	**0.771** (0.756, 0.786)	**0.615** (0.598, 0.632)	**EKPS-CL**	24	**0.827** (0.815, 0.839)	**0.659** (0.647, 0.672)
LR	12	0.701 (0.687, 0.716)	0.471 (0.452, 0.491)	LR	48	0.796 (0.787, 0.806)	0.604 (0.581, 0.628)
RF	12	0.725 (0.714, 0.736)	0.544 (0.523, 0.563)	RF	48	0.805 (0.797, 0.812)	0.613 (0.593, 0.623)
SVM	12	0.691 (0.685, 0.698)	0.504 (0.489, 0.518)	SVM	48	0.787 (0.774, 0.799)	0.610 (0.594, 0.627)
XG-boost	12	0.683 (0.668, 0.699)	0.486 (0.463, 0.509)	XG-boost	48	0.801 (0.791, 0.810)	0.619 (0.598, 0.640)
LSTM	12	0.699 (0.693, 0.706)	0.502 (0.494, 0.511)	LSTM	48	0.806 (0.794, 0.813)	0.636 (0.616, 0.657)
SimCLR	12	0.772 (0.760, 0.782)	0.565 (0.541, 0.584)	SimCLR	48	0.853 (0.843, 0.864)	0.700 (0.689, 0.713)
SupCLR	12	0.783 (0.771, 0.794)	0.590 (0.561, 0.617)	SupCLR	48	0.869 (0.859, 0.879)	0.707 (0.690, 0.726)
EKPS-CL	12	**0.784** (0.771, 0.798)	**0.613** (0.589, 0.637)	**EKPS-CL**	48	**0.872** (0.861, 0.882)	**0.731** (0.711, 0.748)

ACM BCB. Author manuscript; available in PMC 2022 August 10.