Data-Driven Discovery of Feature Groups in Clinical Time Series

Fedor Sergeev[1,2,*] FEDOR.SERGEEV@INF.ETHZ.CH
Manuel Burger[1]
Polina Leshetkina[3]
Vincent Fortuin[2,4,5]
Gunnar Rätsch[1,6,7,8]
Rita Kuznetsova[1,*] RITA.KUZNETSOVA@INF.ETHZ.CH

[1] *Department of Computer Science, ETH Zurich, Switzerland*
[2] *Helmholtz AI, Germany*
[3] *Department of Health Science and Medicine, University of Lucerne, Switzerland*
[4] *TU Munich, Germany*
[5] *Munich Center for Machine Learning, Germany*
[6] *AI Center, ETH Zurich, Switzerland*
[7] *Swiss Institute for Bioinformatics, Zurich, Switzerland*
[8] *Medical Informatics Unit, University Hospital Zurich, Switzerland*

Abstract

Clinical time series data are critical for patient monitoring and predictive modeling. These time series are typically multivariate and often comprise hundreds of heterogeneous features from different data sources. The grouping of features based on similarity and relevance to the prediction task has been shown to enhance the performance of deep learning architectures. However, defining these groups a priori using only semantic knowledge is challenging, even for domain experts. To address this, we propose a novel method that learns feature groups by clustering weights of feature-wise embedding layers. This approach seamlessly integrates into standard supervised training and discovers the groups that directly improve downstream performance on clinically relevant tasks. We demonstrate that our method outperforms static clustering approaches on synthetic data and achieves performance comparable to expert-defined groups on real-world medical data. Moreover, the learned feature groups are clinically interpretable, enabling data-driven discovery of task-relevant relationships between variables.

Keywords: Deep Learning, Healthcare, Time Series, Feature Groups, Clustering.

Data and Code Availability We use the MIMIC-III (Johnson et al., 2016) and HiRID datasets (Faltys et al., 2021), which are freely accessible via the physionet.org platform. We share our code at github.com/ratschlab/feature-group-discovery.

Institutional Review Board (IRB) This research does not require IRB approval.

1. Introduction

In medical research, considering variables through predefined semantic groups rather than individually often helps to reduce data heterogeneity and capture the interactions between features (Kelly and Semsarian, 2009; Meira et al., 2001). This approach is especially relevant for Electronic Health Record (EHR) data, which consists of multivariate time series that describe complex and dynamic patient trajectories. Previous research (Kuznetsova et al., 2023) has shown that grouping EHR variables by their measurement type (vitals, labs, treatments) and organ systems helps improve downstream performance and explainability of deep learning models.

However, when the time series contains dozens or even hundreds of variables, manually defining feature groups that are medically meaningful and improve model performance becomes a challenge even for medical professionals. To address this challenge, we propose a novel algorithm that learns feature groups in a data-driven manner to directly improve model performance during supervised training. Specifically,

* Joint-corresponding authors

© 2025 F. Sergeev, M. Burger, P. Leshetkina, V. Fortuin, G. Rätsch & R. Kuznetsova.

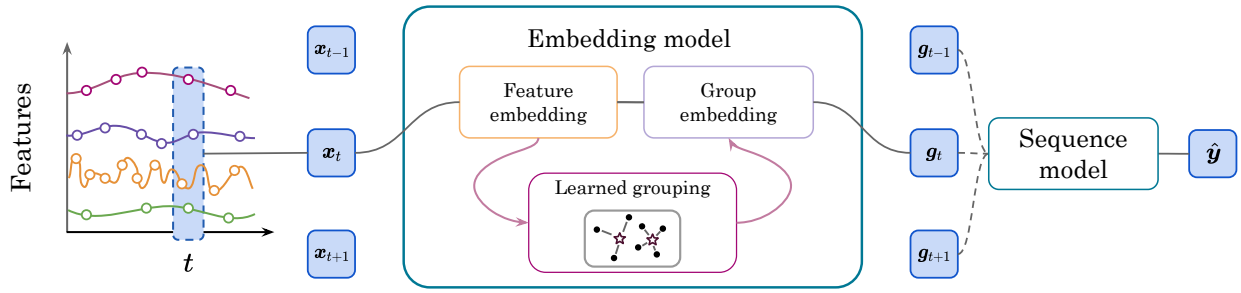


Figure 1: The proposed step-wise embedding model with learned feature groups. Filled and non-filled boxes represent vectors and learnable functions, respectively.

we focus on the *global* feature groups (the same across all patients and time points) (Chormunge and Jena, 2018). These, unlike the sample-wise groups (Masoomi et al., 2020a), are specific to each task and dataset.

We build directly on previous work that has improved deep learning architectures on EHR data (Tomašev et al., 2019, 2021; Yèche et al., 2021; Kuznetsova et al., 2023) and tabular data (Gorishniy et al., 2022, 2023). Specifically, we use an expressive step-wise embedding approach that has shown excellent performance on tabular and time series data by encoding features with neural networks before applying a backbone model. This approach can also be naturally extended to embed features by groups that are defined a priori by domain experts, further improving the performance on EHR data (Kuznetsova et al., 2023). It is based on learning embeddings first for each feature, then for each group, combining group embeddings, and passing them to the sequence model.

We further improve this approach by enabling the model to learn global feature groups during normal supervised training (see Figure 1). Specifically, we characterize each feature by the corresponding feature-wise embedding weights of the step-wise embedding model and cluster them using a centroid-based algorithm. The feature-wise embedding weights are consistent across samples and time steps, guaranteeing that they encode global information about the variables. The weights are updated through gradient descent, which means that the clustering can be seamlessly integrated into the normal training loop. To the best of our knowledge, this is the first work to consider and demonstrate the impact of globally learned feature groups on prediction performance for EHR data.

Contributions (1) We propose a novel method for learning feature groups in time series data by clustering weights of feature-wise embeddings. (2) We show that our method recovers ground-truth feature groups on synthetic data and matches the performance of state-of-the-art models on real Intensive Care Unit (ICU) data without relying on predefined feature groups. (3) We demonstrate that the learned feature groups are novel and interpretable by medical experts, enabling data-driven discovery of task-relevant relationships between clinical variables.

2. Method

We consider a supervised learning problem: given a time series $\boldsymbol{x}$, the task is to predict a label $\boldsymbol{y}$. The time series is represented by a uniform grid: $\boldsymbol{x} = (\boldsymbol{x_1}, \ldots, \boldsymbol{x_T})$, where T is the length of the series.[1]

The step-wise embedding model (see Figure 1) consists of three blocks: the feature embedding model, the group embedding model, and the sequence model (Kuznetsova et al., 2023). The feature embedding model transforms an input time step $\boldsymbol{x}_t$ using a feature-specific linear layer with weights $\boldsymbol{W}_f$ followed by a more expressive nonlinear function ψ:

$$\boldsymbol{h}_t = \psi\left(\{\boldsymbol{W}_f \boldsymbol{x}_{t,f}\}_{f=1}^{F}\right), \tag{1}$$

Here, F is the number of features. The group embedding model then splits the feature embeddings $\boldsymbol{h}_t$

1. Notation is provided in Appendix A.

into K groups, processes them separately, and aggregates them back together into an embedding $\boldsymbol{g}_t$ (see Section 2.1). That embedding is then passed on to the sequence model, which makes the final prediction $\hat{\boldsymbol{y}}$. The model is trained back-to-back using a supervised loss and, in the case of learned groups, a regularization term (see Section 2.3).

The feature groups are defined by a binary membership matrix $\boldsymbol{M} \in \{0,1\}^{F \times K}$, where $\boldsymbol{M}_{f,k} = 1$ if the feature f belongs to the group k, and $\boldsymbol{M}_{f,k} = 0$ otherwise. Although these matrices can be given a priori, we instead propose learning them by clustering the feature-wise embedding weights $\boldsymbol{W}_f$ (see Section 2.2).

2.1. Group Embedding

Before describing the group embedding model, we formally define feature groups. A *feature group* k is a set of feature indices $G_k \subset \{1, \ldots, F\}$. In practice, it is convenient to represent groups using the membership matrix introduced above: $\boldsymbol{M}_k = \{\mathbb{1}(f \in G_k)\}_{f=1}^{F}$. Each feature must belong to at least one group: $\forall f \ \exists k : \ \boldsymbol{M}_{f,k} = 1$. Following Kuznetsova et al. (2023), we initially require each feature to belong to exactly one group: $\forall f \ \exists! \ k : \ \boldsymbol{M}_{f,k} = 1$. For now, we assume that feature groups are given a priori.

We now introduce the group embedding model. First, a *split function* S partitions the feature embeddings $\boldsymbol{h}_t$ into K matrices $\boldsymbol{M}_k \boldsymbol{h}_t$, each corresponding to a group. Then a *group-specific embedding function* ϕ_k independently processed each group (it can be parameterized, for example, by a small Transformer). Finally, an *aggregation function* A combines the resulting group embeddings $\phi_k(\boldsymbol{M}_k \boldsymbol{h}_t)$ into a single step-wise embedding $\boldsymbol{g}_t$. The aggregation function can be implemented as a concatenation, mean, or attention-pooling, with the choice treated as a hyperparameter.

The complete group embedding procedure is given by:

$$\begin{aligned} \boldsymbol{h}'_{t,k} &= S(\boldsymbol{h}_t), && \text{split into groups;} \\ \boldsymbol{g}'_{t,k} &= \phi_k\left(h'_{t,k}\right), && \text{embed the groups;} \\ \boldsymbol{g}_t &= A\left(g'_{t,k}\right), && \text{aggregate the groups.} \end{aligned} \tag{2}$$

The sketch of the architecture described above is shown in Figure 2.

2.2. Learned Groups

Hard groups In the previous section, we assume access to a prior grouping of features. To relax this assumption, we introduce a feature clustering algorithm that produces the membership matrix $\boldsymbol{M}$ used by the splitting function S.

To perform the clustering, the algorithm requires input vectors that characterize each feature. We propose using the feature-wise embedding weights $\{\boldsymbol{W}_f\}$ as these vectors.

However, the dimensions of these matrices may vary between features (e.g., for numerical and one-hot encoded categorical features), making them unsuitable for direct comparison (see Section B.1).

To resolve this, we introduce a *unification function* U that maps (unifies) these matrices $\boldsymbol{W}_f = \left[\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{C_f}, \boldsymbol{w}_{\text{bias}}\right]$, where C_f is the size of encoding for feature f. Specifically, U combines the weight columns and concatenates them to the bias column. The function U can then be written as:

$$U\left(\boldsymbol{W}_f\right) = \left[\boldsymbol{w}_{\text{bias}}^{\top}, \ \text{Combine}\left(\{\boldsymbol{w}_i\}_{i=1}^{C_f}\right)^{\top}\right]^{\top}. \tag{3}$$

The combination function can sum the weight columns, take their mean, or ignore them entirely (producing an empty vector). The choice of this operation is treated as a hyperparameter (see Appendix B.2).

Now, we can use any clustering algorithm on the set of vectors $\{U(\boldsymbol{W}_f)\}_{f=1}^{F}$ that characterize the features (see Figure 2). The algorithm then outputs the membership matrices $\{\boldsymbol{M}_k\}_{k=1}^{K}$.

In this work, we consider centroid-based clustering algorithms: *K-means*, *Fuzzy K-means*, and Gaussian mixture model (*GMM*) clustering algorithms (Hartigan and Wong, 1979; Bezdek et al., 1984; Dempster et al., 1977). We focus on these algorithms because we find that controlling the dynamics of the centroid μ_k update is a useful form of regularization (see Section 4 for further discussion).

Soft groups In the previous sections, following Kuznetsova et al. (2023), we assumed a hard clustering, where each feature belongs to exactly one cluster. However, in real data, it is natural for a variable to interact with multiple groups of features and, therefore, to belong to multiple clusters. For example, in a medical setting, variables that describe body fluid levels can be relevant to both the circulatory and renal systems.

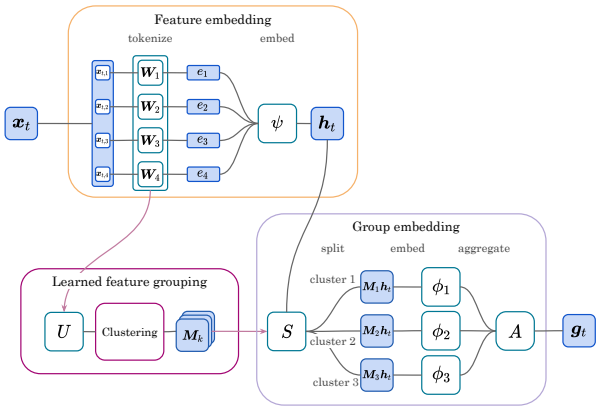


Figure 2: Learned feature clustering scheme.

To account for this, we relax the hard clustering assumption by introducing a soft membership model. Instead of producing an indicator assignment vector for each feature, the algorithm produces a vector $\boldsymbol{p}_f$ that scores the "probability" of feature f belonging to each group.

$$\boldsymbol{p}_f^k = \mathbb{P}(\text{feature } f \text{ is assigned to cluster } k),$$

where the score values are normalized but are not required to sum to one. The cluster membership matrix is then defined as

$$\boldsymbol{M}_{f,k} = \mathbb{1}\left(\frac{\boldsymbol{p}_f^k}{\max_l \boldsymbol{p}_f^l} > \Delta_k\right), \tag{4}$$

where $\Delta_k \in [0, 1)$ is the threshold on the membership score for assigning a feature to the cluster k. Here, division of the score vector by its maximum element prevents the clusters from being empty. Note that hard clustering is essentially an edge case of soft clustering when $\Delta \to 1$.

The thresholds Δ_k are treated as hyperparameters. For simplicity, we set them equal to Δ for all k. We discuss practical implementation and other options for the membership function in Appendix B.5.

2.3. Training and Regularization

The resulting model is trained using gradient descent with an additional update to the clustering (see Algorithm 1). Specifically, every feature f is assigned to a specific cluster k based on $\boldsymbol{W}_f$. During the backpropagation phase, the weight matrices $\{\boldsymbol{W}_f\}$ are updated based on the loss gradient. The feature clustering can then be periodically recomputed independently of the backpropagation. For example, every P iterations, where P is a hyperparameter. For simplicity, we set $P = 1$.

Algorithm 1: Training learned feature groups

Input: Num. of groups K, reg. coef. λ, EMA coef. α
Initialize: Feature groups $\{\boldsymbol{M}_k\}$, centroids $\{\boldsymbol{\mu}_k\}$
while *training* **do**
  for *each* t **do**
    $\boldsymbol{h}_t \leftarrow$ Feature embedding$(\boldsymbol{x}_t)$ // (1)
    $\boldsymbol{g}_t \leftarrow$ Group embedding $(\boldsymbol{h}_t, \{\boldsymbol{M}_k\})$ // (2)
  end
  $\hat{\boldsymbol{y}} \leftarrow$ Sequence model$(\boldsymbol{g})$
  $\mathcal{L} \leftarrow \mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}) + \lambda \mathcal{L}_{\text{reg}}(\{\boldsymbol{\mu}_k\}, \{\boldsymbol{W}_f\})$ // (6)
  Gradient step
  $\{\boldsymbol{M}_k\} \leftarrow \text{Clustering}_{\{\boldsymbol{\mu}_k\}}(\{U(\boldsymbol{W}_f)\})$ // (3)
  if $\{\boldsymbol{M}_k\}$ *was updated* **then**
    $\boldsymbol{\mu}_k \leftarrow \text{EMA}_\alpha(\boldsymbol{\mu}_k, \boldsymbol{\mu}'_k)$ // (7)
    $\{\boldsymbol{M}_k\} \leftarrow \text{Clustering}_{\{\boldsymbol{\mu}_k\}}(\{U(\boldsymbol{W}_f)\})$ // (3)
  end
end

Since $\{\boldsymbol{W}_f\}$ are updated during training, clusters could fail to form if weight matrices change much from one epoch to another. To promote the formation of clusters within the latent feature space and improve training stability, we introduce a *regularization loss term* $\mathcal{L}_{\text{reg}}$. Denoting cluster centroids (or means in the GMM case) $\{\boldsymbol{\mu}_k\}_{k=1}^K$:

$$\mathcal{L}_{\text{reg}}\left(\{U(\boldsymbol{W}_f)\}_{f=1}^F\right) = \frac{\frac{1}{K}\sum_{k=1}^K \mathcal{L}_k^{\text{intra}}}{\frac{1}{K}\sum_{k=1}^K \mathcal{L}_k^{\text{inter}}}, \tag{5}$$

Here $\mathcal{L}^{\text{inter}}$ is an *intercluster loss* and $\mathcal{L}^{\text{intra}}$ is an *intracluster loss* defined as

$$\mathcal{L}_k^{\text{intra}} = \sum_{i \in G_k} \|U(\boldsymbol{W}_i) - \boldsymbol{\mu}_k\|, \quad \mathcal{L}_k^{\text{inter}} = \sum_{k'=1}^K \|\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'}\|.$$

The former is responsible for decreasing the distances between features within the assigned cluster ("tightening" the clusters). The latter is responsible for increasing the distances between different clusters ("pushing" them apart).

This regularization term is important for ensuring the establishment of a robust clustering structure, especially in the early stages of training. Note that the gradient is only taken with respect to $\boldsymbol{W}_f$, so this function is indeed differentiable and only affects the feature embedding function defined in Equation (1).

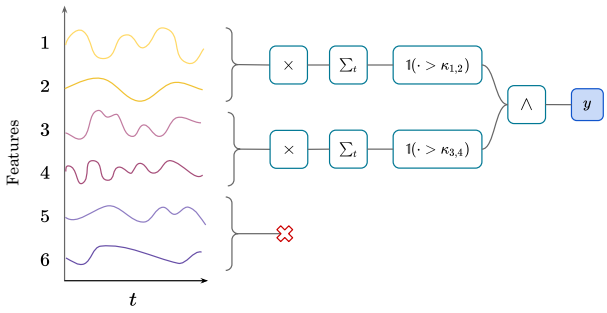


Figure 3: Synthetic data labeling procedure. Feature values are sampled from Gaussian Processes with various parameters.

The final loss then takes the form

$$\mathcal{L} = \mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{y}}) + \lambda \mathcal{L}_{\text{reg}} \left(\{\boldsymbol{\mu}_k\}_{k=1}^{K}, \{\boldsymbol{W}_f\}_{f=1}^{F} \right), \quad (6)$$

where $\mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{y}})$ is the supervision loss and $\lambda \in [0, 1)$ is the weight of the clustering loss. Combining supervised task objective and feature group learning enables grouping of features based on their mutual relationships and relevance to the task label.

To further improve training stability, we modify the update of $\boldsymbol{\mu}$ to take into account the previous centroid position by calculating an *expected moving average* (EMA). This provides an additional degree of control over the dynamics of centroid movement.

Specifically, for K-means algorithms, given the centroids from the previous iteration $\boldsymbol{\mu}_k$ and the newly calculated centroids $\boldsymbol{\mu}'_k$, the update is

$$\boldsymbol{\mu}_k = \alpha \boldsymbol{\mu}_k + (1 - \alpha) \boldsymbol{\mu}'_k, \quad (7)$$

where $\alpha \in [0, 1)$ is the *decay parameter*.

Similarly, for GMMs we can compute an "EMA" between the Gaussian distributions using moment matching (see Appendix B.4 for a discussion),

$$\begin{aligned} \boldsymbol{\mu}_k &= \alpha \boldsymbol{\mu}_k + (1 - \alpha) \boldsymbol{\mu}'_k, \\ \boldsymbol{\Sigma}_k &= \alpha \boldsymbol{\Sigma}_k + (1 - \alpha) \boldsymbol{\Sigma}'_k + \\ &\quad + \alpha(1 - \alpha)(\boldsymbol{\mu}_k - \boldsymbol{\mu}'_k) \times (\boldsymbol{\mu}_k - \boldsymbol{\mu}'_k)^T. \end{aligned} \quad (8)$$

3. Experiments

3.1. Synthetic Data

3.1.1. Setup

First, we study the performance of our method on a synthetic dataset. While on real data, the target underlying feature groups are not known, we design the synthetic dataset where the true grouping is fixed. The dataset is simple, enabling comparison to usual clustering methods that use simple aggregated information to characterize the features.

Table 1: Clustering quality on synthetic data. Means and standard deviations are computed over 5 runs. Best non-oracle performance is highlighted in bold. Dynamic clustering outperforms static clustering, which fails to learn ground-truth labels.

Algorithm / Input	ARI (↑)	NMI (↑)	Silhouette
Reference models			
Random	-0.00 ± 0.57	0.40 ± 0.37	-0.001 ± 0.002
Oracle	1.00 ± 0.00	1.00 ± 0.00	-0.003 ± 0.000
Static K-means			
$\mathcal{D}_{\text{flat}}$	-0.24 ± 0.10	0.38 ± 0.05	0.002 ± 0.003
$\mathcal{D}_{\text{time mean}}$	-0.14 ± 0.20	0.39 ± 0.11	0.004 ± 0.005
$\mathcal{D}_{\text{sample mean}}$	-0.03 ± 0.06	0.43 ± 0.04	**0.110** ± 0.060
$\mathcal{D}_{\text{full mean}}$	0.04 ± 0.06	0.49 ± 0.05	-0.001 ± 0.002
Dynamic K-means			
$\mathcal{D}$	**0.23** ± 0.51	**0.59** ± 0.29	-0.001 ± 0.002

To study the difference between these data representations, we use the same clustering algorithm for all settings. For simplicity, we choose K-means as other clustering methods can be seen as its generalization.

We generate a synthetic multivariate time series dataset with 6 features of length 20, where the feature values are sampled from Gaussian processes (GP) with individual parameters (see Figure 3). We define the ground-truth feature groups $\{\{1, 2\}, \{3, 4\}, \{5, 6\}\}$. Using these groups, we assign a binary label to each time series as follows. First, we sum the values of the feature products of 1, 2 and 3, 4 over all time steps. Then we calculate the indicator function to determine whether the aggregated values are greater than the thresholds $\kappa_{1,2}$ and $\kappa_{3,4}$. The thresholds are chosen as medians of the respective sums over the dataset. Finally, the two indicators are combined into a single label y via a logical "and" operation. The features 5 and 6 form a group, as they both do not correlate with the label (see Appendix C.2 for further details).

Classical clustering algorithms require an input vector that characterizes the features and their relationship to the sample label y. Therefore, the input dataset $\mathcal{D}$ must be converted to a suitable format. We

construct the new dataset as a concatenation of the label values y with a vector representing the feature values. For the latter, we consider the following options: (1) concatenation of all feature values across all samples i and time steps t ($\mathcal{D}_{\text{flat}}$); (2) concatenation of the feature values averaged over time ($\mathcal{D}_{\text{time mean}}$); (3) concatenation of the feature values averaged over the samples ($\mathcal{D}_{\text{sample mean}}$); (4) concatenation of the feature values averaged over both samples and time ($\mathcal{D}_{\text{full mean}}$). The formal definitions of these transformations are provided in Appendix C.2.

A more sophisticated approach would involve training a sequence model to capture the interaction between features and the target label. This is precisely what our proposed algorithm does, without requiring any modification of the input dataset $\mathcal{D}$.

We compare the ability of this static approach and our proposed dynamic algorithm to recover the target feature groups. For reference, we also report results for random clustering and an oracle.

Performance is evaluated using the Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), and the silhouette score. ARI and NMI are metrics that measure how close the learned clustering is to the true feature groups (Luecken et al., 2022; Hassan et al., 2021). They are in range [-1,1] and [0,1] respectively, with higher values indicating higher similarity. ARI measures how many permutations it would take to transform one into another. NMI represents the mutual information between true and predicted clusterings. The silhouette score, unlike ARI and NMI, does not rely on true feature groups. Instead, it measures how well the clusters are formed in a geometric sense using distances in the embedding space (see Appendix C.1 for details).

3.1.2. Results

We present the results in Table 1. We see that the static K-means fails to recover the feature groups for all data representations, performing no better, and often worse, than random clustering. In contrast, dynamic K-means can recover the correct feature groups for at least some initializations.

We also notice that the silhouette score for the oracle is lower than for the random clustering and most static K-means assignments. This indicates that optimizing an internal clustering metric, such as the silhouette score, does not necessarily correspond to recovering the true feature groups. This finding highlights the importance of using a downstream metric that ensures that the learned groups actually benefit the target application.

3.2. Medical Data

3.2.1. Setup

Next, we evaluate our method's ability to improve downstream performance on real-world data. We consider two widely used Intensive Care Unit (ICU) datasets: HiRID (Faltys et al., 2021) and MIMIC-III (Johnson et al., 2016). We follow the pre-processing and labeling protocol introduced by Yèche et al. (2021) and subsequently adopted by Yèche et al. (2024); Kuznetsova et al. (2023). The technical details and the full training setup are provided in Appendix C.3.2.

We consider mortality and circulatory failure prediction on HiRID, and mortality and decompensation prediction on MIMIC-III. Mortality prediction (*Mort*) is an offline binary classification task: given a time series, the model predicts whether the patient will survive at the end of their stay. Circulatory failure (*Circ*) and decompensation (*Decomp*) are early event prediction tasks (online binary classification): at each time step, the model predicts whether an event will occur in the next few hours (see Appendix C.3.1 for more details).

Given the strong class imbalance across all tasks, we evaluate performance using the Area Under the Precision-Recall Curve (AUPRC). We also report the Area Under the Receiver Operator Curve (AUROC), although it has been shown to tend to saturate in this setting (Saito and Rehmsmeier, 2015), so we prioritize AUPRC (see Appendix C.1 for more details).

We compare models that learn feature groups with models that use groups defined a priori. Following Kuznetsova et al. (2023), previous groups are defined based on (1) measurement type (e.g., vital signs, laboratory tests) and (2) the organ system to which the variables belong (e.g., circulatory, nervous). We also report reference results for models that do not use any feature grouping, with numbers taken directly from Kuznetsova et al. (2023), where the FT-Transformer (Gorishniy et al., 2023) is used as the base architecture. Specific assignments are listed in Appendix C.3.5.

3.2.2. Results

Downstream performance The results of different clustering algorithms along with models us-

Table 2: **Performance benchmark for different grouping schemes** (AUPRC ↑) on online and offline classification tasks using ICU datasets. Mean and standard deviation are reported over three training runs with different random seeds. Best results are highlighted in bold. Reference results for prior groups (Prior) and models without feature grouping (None) are taken from Kuznetsova et al. (2023). Learned groupings perform comparably to expert-defined ones.

Feature Grouping			HIRID		MIMIC-III	
Type	Membership	Algorithm	Circ	Mort	Decomp	Mort
None	–	–	0.388 ± 0.006	0.605 ± 0.006	**0.387** ± 0.003	0.512 ± 0.008
Prior	Hard	Type	**0.402** ± 0.004	**0.627** ± 0.019	0.380 ± 0.004	0.521 ± 0.001
Prior	Hard	Organ	**0.406** ± 0.004	**0.623** ± 0.012	0.374 ± 0.001	**0.526** ± 0.006
Learned	Hard	K-means	0.393 ± 0.006	**0.631** ± 0.009	0.381 ± 0.002	**0.525** ± 0.006
Learned	Hard	Fuzzy K-means	0.382 ± 0.010	**0.629** ± 0.012	**0.386** ± 0.007	**0.523** ± 0.003
Learned	Hard	GMM	0.399 ± 0.001	**0.625** ± 0.006	0.381 ± 0.001	**0.519** ± 0.011
Learned	Soft	Fuzzy K-means	0.395 ± 0.002	0.615 ± 0.001	0.377 ± 0.006	**0.521** ± 0.009
Learned	Soft	GMM	**0.403** ± 0.004	0.610 ± 0.022	0.380 ± 0.003	**0.520** ± 0.006

ing prior groups (Prior) and models without feature grouping (None) are reported in Table 2. Our models generally outperform architectures without grouping and match the best results achieved by using the prior grouping in Kuznetsova et al. (2023). Additional results, including AUROC and p-values, are reported in Appendix C.3.3.

Interpretation of learned groups The learned groups, analyzed in detail with the help of a medical doctor in Appendix C.3.4, are generally clinically explainable. Models trained on the same task tend to discover related groups, frequently capturing either the baseline state of the patient or specific types of shock. However, larger clusters sometimes contain heterogeneous sets of variables, which reduces explainability and suggests that controlling for cluster size may be beneficial. These findings indicate potential for refining expert-defined feature groups by tailoring them to specific prediction tasks and patient cohorts.

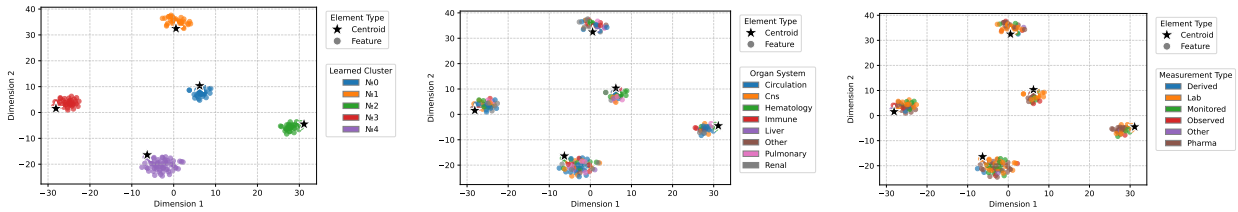
For the decompensation prediction on MIMIC-III, we observe that models without grouping outperform those with expert-defined groups. Examining the clustering dynamics during training (Figure 12), we find that the model learns to assign all features to a single group. This behavior suggests that the method can correct a suboptimal initialization, rediscovering the most suitable grouping strategy (see Appendix B.6 for further discussion). We discuss the possible reason behind the no-grouping outperforming algorithms with grouping on this task in Appendix C.3.4.

At the same time, the feature groups discovered in other tasks are novel. In Figure 4, we visualize the embedding space of the model trained on HiRID for mortality prediction using t-SNE (van der Maaten and Hinton, 2008), coloring the features according to the learned and expert-defined groupings. The learned clusters are well-formed, yet distinct from the expert-defined groups, indicating that the method uncovers new, task-relevant structures.

Ablations To better understand how the model hyperparameters affect its performance, we perform ablation studies and report their results in Appendix C.3.6. They show that regularization and EMA improve performance on several tasks. However, there is no single best choice for the initialization method, number of clusters, unification function, or merge function, as the optimal configuration varies between tasks.

4. Discussion and Limitations

Our method has shown comparable downstream performance to the a priori defined feature groups. This indicates that the feature groups proposed by Kuznetsova et al. (2023) are indeed a strong baseline. However, our method is able to match this baseline while providing greater flexibility. For example, on new data or tasks, it could automatically discover



(a) By learned cluster (b) By organ system (c) By measurement type

Figure 4: Embedding space of learned features colored according to learned and prior groupings (K-means, HiRID mortality). The embedding space is visualized using t-SNE of Euclidean distances between centroids and features. The model produces novel feature clusters that differ from prior groupings.

feature groups that lead to strong downstream performance, which is not guaranteed for prior groupings.

This flexibility comes at the cost of increased complexity in two respects, (1) computational cost and (2) additional hyperparameters. The computational cost arises from feature clustering, but since this step does not require GPU resources, we found that training times with prior and learned clustering were comparable in practice (see Appendix C.3.6). The additional hyperparameters include the number of clusters, the EMA coefficient, and the type and strength of regularization. However, given that base architectures already involve dozens of hyperparameters and require extensive tuning (Yèche et al., 2021; Kuznetsova et al., 2023), the additional hyperparameters might not pose a significant limitation in practice.

In this work, we focused on centroid-based clustering algorithms, which allow additional regularization through the EMA of centroids. We did not explore centroid-less methods such as DBSCAN (Hahsler et al., 2019) and Gumbel-Softmax NN-based approaches (Imrie et al., 2022; Jang et al., 2016). Since these methods lack centroid regularization, they may face challenges in dynamic training settings. Similarly, we have not considered updating centroids using gradients, as this might make the clustering less stable due to the stochasticity of the gradient descent. We leave the investigation of these methods to future work.

Our model currently relies on clustering algorithms with a fixed number of clusters, specified as a hyperparameter. Although in practice models may use fewer clusters (e.g., K-means for decompensation prediction), automatic discovery of the optimal number of clusters remains an open direction for future research (we discuss this problem in more depth in Section B.7 and provide initial results for the Elbow method in Section C.3.3).

Finally, we have limited the scope of our experiments to in-distribution evaluation. In the real world, generalization across hospitals is a critical challenge (Moor et al., 2023; Burger et al., 2024, 2025; Van De Water et al., 2023). Our model is naturally adaptable to such settings and could potentially reveal how feature groups vary across institutions, thereby providing insights into distribution shifts.

5. Related Work

ML for ICU Large EHR datasets have enabled the rapid development of machine learning models for ICU time series. Harutyunyan et al. (2019); Hyland et al. (2020); Yèche et al. (2021) formalized clinical prediction tasks and developed strong tree-based and deep learning architectures for MIMIC-III and HiRID datasets. Yèche et al. (2023); Yèche et al. (2024); Kuznetsova et al. (2023) further improved deep learning models by modifying the loss function and architecture. We build directly on this line of research, adopting the same datasets and task definitions.

Feature embedding in time series Out-of-the-box deep neural networks often struggle to match the performance of tree-based architectures on tabular data. To address this shortcoming, Gorishniy et al. (2022) proposed various strategies for feature embedding and incorporated them into ResNet (He et al., 2016) and Transformer (Vaswani et al., 2017) models. Tomašev et al. (2021, 2019); Kuznetsova et al.

(2023) adapted these innovations to ICU time series with various embedding architectures.

Feature grouping Predefined feature groups have been shown to improve the downstream performance and explainability of time series models. Kuznetsova et al. (2023); Tomašev et al. (2019, 2021) employed expert-defined feature groups to improve downstream performance on ICU time series data. Swamy et al. (2024) used predefined feature groups to design an inherently interpretable neural network. We extend these ideas by *learning* feature groups to directly improve downstream performance while enhancing model explainability.

Several works have explored data-driven feature grouping before, but with key differences from our approach. Masoomi et al. (2020b) considered *local* (or instance-wise) feature grouping, while we focus on *global* feature groups that could help predictive modeling for a set clinical task in all patients. Imrie et al. (2022) also considered global feature grouping, learning feature groups for static data using an ensemble-based method. Our method is tailored for time series and learns groups by clustering embedding weights directly within a deep learning architecture.

In a related context, the same clustering algorithms have been applied to time series in the context of sample grouping (Ma et al., 2019), while our focus is on grouping features. Similarly, interdependencies between features can be expressed via attention weights (Ma et al., 2020), while our method produces explicit and global groups. Finally, feature grouping has also been used to achieve other goals, such as improving the computational efficiency of the attention mechanism (Roy et al., 2021; Vyas et al., 2020), tailoring it to specific downstream tasks (Li et al., 2021), or phenotyping patients via tensor decomposition (Becker et al., 2023).

6. Conclusion

In this work, we introduce a novel method for learning feature groups in clinical time series by clustering weights of feature-wise embeddings. Our experiments show that this dynamic model is capable of recovering ground-truth feature groups on synthetic data. On real-world ICU datasets, our method produces novel feature groups that match the performance of expert-defined groups. We show that the model produces interpretable clusters and is capable of rediscovering the best-known groupings. Looking ahead, this framework could be used to explore feature groups across tasks and hospitals more broadly, offering valuable data-driven insights for building more robust and explainable models.

Acknowledgments

We thank the anonymous reviewers for their helpful suggestions for our paper. We thank researchers, patients, and clinical staff for creating and sharing the medical datasets used in this work. Computational data analysis was performed at Leonhard Med (https://sis.id.ethz.ch/services/sensitiveresearchdata/), a secure, trusted research environment at ETH Zurich. Fedor Sergeev was supported by grant #902 of the Strategic Focus Area "Personalized Health and Related Technologies (PHRT)" of the ETH Domain (Swiss Federal Institutes of Technology). Vincent Fortuin was supported by a Branco Weiss Fellowship.

References

Martial Agueh and Guillaume Carlier. Barycenters in the Wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011.

Daniel Alspach and Harold Sorenson. Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE transactions on automatic control*, 17(4):439–448, 2003.

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

Florian Becker, Age K Smilde, and Evrim Acar. Unsupervised EHR-based phenotyping via matrix and tensor decompositions. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13 (4):e1494, 2023.

James C. Bezdek, Robert Ehrlich, and William Full. FCM: The fuzzy c-means clustering algorithm. *Computers & geosciences*, 10(2-3):191–203, 1984.

Manuel Burger, Fedor Sergeev, Malte Londschien, Daphné Chopard, Hugo Yèche, Eike Gerdes, Polina Leshetkina, Alexander Morgenroth, Zeynep Babür, Jasmina Bogojeska, et al. Towards foundation models for critical care time series. *arXiv preprint arXiv:2411.16346*, 2024.

Manuel Burger, Daphné Chopard, Malte Londschien, Fedor Sergeev, Hugo Yèche, Rita Kuznetsova, Martin Faltys, Eike Gerdes, Polina Leshetkina, Peter Bühlmann, et al. A foundation model for intensive care unlocking generalization across tasks and domains at scale. *medRxiv*, pages 2025–07, 2025.

Yanshuai Cao and David J. Fleet. Generalized product of experts for automatic and principled fusion of Gaussian process predictions. *arXiv preprint arXiv:1410.7827*, 2014.

Davide Chicco and Giuseppe Jurman. The matthews correlation coefficient (mcc) should replace the roc auc as the standard metric for assessing binary classification. *BioData Mining*, 16(1):4, 2023.

Smita Chormunge and Sudarson Jena. Correlation based feature selection with clustering for high dimensional data. *Journal of Electrical Systems and Information Technology*, 5(3):542–549, 2018.

Marco Cuturi and Arnaud Doucet. Fast computation of Wasserstein barycenters. In *International conference on machine learning*, pages 685–693. PMLR, 2014.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977.

P Kingma Diederik. Adam: A method for stochastic optimization. *(No Title)*, 2014.

M. Faltys, M. Zimmermann, X. Lyu, M. Hüser, S. Hyland, G. Rätsch, and T. Merz. HiRID, a high time-resolution icu dataset (version 1.1.1). *PhysioNet*, 2021. doi: 10.13026/nkwc-js72. URL https://doi.org/10.13026/nkwc-js72.

Thomas S Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.

Yury Gorishniy, Ivan Rubachev, and Artem Babenko. On embeddings for numerical features in tabular deep learning. *Advances in Neural Information Processing Systems*, 35:24991–25004, 2022.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data, 2023. URL https://arxiv.org/abs/2106.11959.

Cyril Goutte and Eric Gaussier. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *European conference on information retrieval*, pages 345–359. Springer, 2005.

Michael Hahsler, Matthew Piekenbrock, and Derek Doran. dbscan: Fast density-based clustering with R. *Journal of Statistical Software*, 91(1):1–30, 2019. doi: 10.18637/jss.v091.i01.

John A. Hartigan and Manchek A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108, 1979.

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):1–18, 2019.

Bryar A Hassan, Tarik A Rashid, and Hozan K Hamarashid. A novel cluster detection of covid-19 patients and medical disease conditions using improved evolutionary clustering algorithm star. *Computers in biology and medicine*, 138:104866, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

Stephanie L Hyland, Martin Faltys, Matthias Hüser, Xinrui Lyu, Thomas Gumbsch, Cristóbal Esteban, Christian Bock, Max Horn, Michael Moor, Bastian Rieck, et al. Early prediction of circulatory failure in the intensive care unit using machine learning. *Nature medicine*, 26(3):364–373, 2020.

Fergus Imrie, Alexander Norcliffe, Pietro Liò, and Mihaela van der Schaar. Composite feature selection using deep ensembles. *Advances in Neural Information Processing Systems*, 35:36142–36160, 2022.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Alistair Johnson, Tom Pollard, and Mark Roger. MIMIC-III Clinical Database, 2016. URL https://physionet.org/content/mimiciii/1.4/.

Matthew Kelly and Christopher Semsarian. Multiple mutations in genetic cardiovascular disease: a marker of disease severity? *Circulation: Cardiovascular Genetics*, 2(2):182–190, 2009.

Rita Kuznetsova, Alizée Pace, Manuel Burger, Hugo Yèche, and Gunnar Rätsch. On the importance of step-wise embeddings for heterogeneous clinical time-series. In *Machine Learning for Health (ML4H)*, pages 268–291. PMLR, 2023.

Tilman Lange, Volker Roth, Mikio L Braun, and Joachim M Buhmann. Stability-based validation of clustering solutions. *Neural computation*, 16(6): 1299–1323, 2004.

Shuaicheng Li, Qianggang Cao, Lingbo Liu, Kunlin Yang, Shinan Liu, Jun Hou, and Shuai Yi. Groupformer: Group activity recognition with clustered spatial-temporal transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13668–13677, 2021.

Malte D Luecken, Maren Büttner, Kridsadakorn Chaichoompu, Anna Danese, Marta Interlandi, Michaela F Müller, Daniel C Strobl, Luke Zappia, Martin Dugas, Maria Colomé-Tatché, et al. Benchmarking atlas-level data integration in single-cell genomics. *Nature methods*, 19(1):41–50, 2022.

Liantao Ma, Chaohe Zhang, Yasha Wang, Wenjie Ruan, Jiangtao Wang, Wen Tang, Xinyu Ma, Xin Gao, and Junyi Gao. Concare: Personalized clinical feature embedding via capturing the healthcare context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 833–840, 2020.

Qianli Ma, Jiawei Zheng, Sen Li, and Gary W Cottrell. Learning representations for time series clustering. *Advances in neural information processing systems*, 32, 2019.

Aria Masoomi, Chieh Wu, Tingting Zhao, Zifeng Wang, Peter Castaldi, and Jennifer Dy. Instance-wise feature grouping. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 13374–13386. Curran Associates, Inc., 2020a. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/9b10a919ddeb07e103dc05ff523afe38-Paper.pdf.

Aria Masoomi, Chieh Wu, Tingting Zhao, Zifeng Wang, Peter Castaldi, and Jennifer Dy. Instance-wise feature grouping. *Advances in Neural Information Processing Systems*, 33:13374–13386, 2020b.

Lisiane B. Meira, Antonio MC. Reis, David L. Cheo, Dorit Nahari, Dennis K. Burns, and Errol C. Friedberg. Cancer predisposition in mutant mice defective in multiple genetic pathways: uncovering important genetic interactions. *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, 477(1-2):51–58, 2001.

Michael Moor, Oishi Banerjee, Zahra Shakeri Hossein Abad, Harlan M Krumholz, Jure Leskovec, Eric J Topol, and Pranav Rajpurkar. Foundation models for generalist medical artificial intelligence. *Nature*, 616(7956):259–265, 2023.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Peter J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9: 53–68, 2021.

Andrew R. Runnalls. Kullback-Leibler approach to Gaussian mixture reduction. *IEEE Transactions on Aerospace and Electronic Systems*, 43(3):989–999, 2007.

Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.

Student. The probable error of a mean. *Biometrika*, pages 1–25, 1908.

Vinitra Swamy, Syrielle Montariol, Julian Blackwell, Jibril Frej, Martin Jaggi, and Tanja Käser. InterpretCC: intrinsic user-centric interpretability through global mixture of experts. *Network*, 2:G1, 2024.

Robert L Thorndike. Who belongs in the family? *Psychometrika*, 18(4):267–276, 1953.

Nenad Tomašev, Xavier Glorot, Jack W Rae, Michal Zielinski, Harry Askham, Andre Saraiva, Anne Mottram, Clemens Meyer, Suman Ravuri, Ivan Protsyuk, et al. A clinically applicable approach to continuous prediction of future acute kidney injury. *Nature*, 572(7767):116–119, 2019.

Nenad Tomašev, Natalie Harris, Sebastien Baur, Anne Mottram, Xavier Glorot, Jack W Rae, Michal Zielinski, Harry Askham, Andre Saraiva, Valerio Magliulo, et al. Use of deep learning to develop continuous-risk models for adverse event prediction from electronic health records. *Nature protocols*, 16(6):2765–2787, 2021.

Robin Van De Water, Hendrik Schmidt, Paul Elbers, Patrick Thoral, Bert Arnrich, and Patrick Rockenschaub. Yet another ICU benchmark: A flexible multi-center framework for clinical ML. *arXiv preprint arXiv:2306.05109*, 2023.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

NX. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: Variants. *Properties, normalization and correction for chance*, 18, 2009.

Apoorv Vyas, Angelos Katharopoulos, and François Fleuret. Fast transformers with clustered attention. *Advances in Neural Information Processing Systems*, 33:21665–21674, 2020.

Shashank Yadav and Vignesh Subbian. Failure modes of time series interpretability algorithms for critical care applications and potential solutions. *arXiv preprint arXiv:2506.19035*, 2025.

Hugo Yèche, Rita Kuznetsova, Marc Zimmermann, Matthias Hüser, Xinrui Lyu, Martin Faltys, and Gunnar Rätsch. HiRID-ICU-benchmark — a comprehensive machine learning benchmark on high-resolution ICU data. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

Hugo Yèche, Alizée Pace, Gunnar Ratsch, and Rita Kuznetsova. Temporal label smoothing for early event prediction. In *International Conference on Machine Learning*, pages 39913–39938. PMLR, 2023.

Hugo Yèche, Manuel Burger, Dinara Veshchezerova, and Gunnar Rätsch. Dynamic survival analysis for early event prediction, 2024. URL https://arxiv.org/abs/2403.12818.

Appendix A. Notation

- Data
  - $\boldsymbol{x} \in \mathbb{R}^{T \times \sum_{f=1}^{F} C_f}$ — time series
  - $\boldsymbol{x}_t$ — time step $t \in \{1, \ldots, T\}$
  - $\boldsymbol{x}_{t,f}$ — feature $f \in \{1, \ldots, F\}$ at time step t
  - $\boldsymbol{y}$ — true label, either $\boldsymbol{y} \in \{0, 1\}$ for binary classification or $\boldsymbol{y} \in \{0, 1\}^T$ for early event prediction
  - $\hat{\boldsymbol{y}}$ — predicted label
  - T — length of the time series
  - F — number of features
  - N_{num} — number of numerical features
  - N_{cat} — number of categorical features
  - C_f — size of feature f encoding, equals to one for numerical and to the number of classes for categorical features
  - D — size of the dataset
- Embedding architecture
  - H — token size
  - $\boldsymbol{W}_f \in \mathbb{R}^{C_f \times H}$ — tokenization matrix
  - $\boldsymbol{w}_i$ — i-th column of $\boldsymbol{W}_f$
  - ψ — nonlinear feature embedding function
  - K — number of feature groups
  - $G_k \in 2^F$ — set of indices of features belonging to the group $k \in \{1, \ldots, K\}$
  - $\boldsymbol{M}$ — feature-group membership matrix; for hard clustering $\boldsymbol{M} \in \{0, 1\}^{F,K}$ where $\boldsymbol{M}_{k,f} = 1$ iff $f \in G_i$, for soft clustering $\boldsymbol{M} \in \mathbb{R}^{F,K}$
  - S — splitting function
  - ϕ_k — group embedding function for group k
  - A — aggregation function
  - $\boldsymbol{g}_t$ — final embedding
- Learned grouping
  - U — function converting feature embedding matrices $\boldsymbol{W}_f$ to the same dimensionality
  - $\boldsymbol{\mu}$ — centroid or GMM
  - Σ — GMM covariance
  - $\mathcal{L}_{\text{reg}}$ — cluster regularization loss term
  - $\mathcal{L}^{\text{inter}}$ — intercluster loss, "tightens" the clusters
  - $\mathcal{L}^{\text{intra}}$ — intracluster loss, "pushes" the clusters apart
  - λ — regularization loss term coefficient
  - α — centroid moving decay parameter
  - $\boldsymbol{p}_f \in [0, 1]^K$— cluster membership score vector for feature f
  - Δ – cluster membership threshold

Appendix B. Method Details

B.1. Categorical and Numerical Features

In the medical time series we consider, each feature $\boldsymbol{x}_{t,f}$ can be categorical or numerical. The categorical features are considered to be one hot encoded: $\boldsymbol{x}_{t,f} \in \mathbb{1}^{C_f}$, where C_f is the number of categories for feature f. The numerical features are simply represented by real numbers: $\boldsymbol{x}_{t,f} \in \mathbb{R}^{C_f}$, $C_f = 1$. We denote the number of categorical and numerical features N_{cat} and N_{num}, respectively. Without loss of generality, we will assume that numerical features are indexed $\{1, \ldots, N_{cat}\}$. Then we can write $\boldsymbol{x} \in \mathbb{R}^{T \times (N_{\text{num}} + \sum_{f=1}^{N_{\text{cat}}} C_f)}$.

The embedding matrices then have the following dimensions $\boldsymbol{W}_f \in \mathbb{R}^{(C_f+1) \times H}$ (+1 for the bias term). For numerical features, the dimensions of $\boldsymbol{W}_f$ are the same for any f: the matrix contains two columns for the weight and the bias term, respectively. For categorical variables, $\{\boldsymbol{W}_f\}$ has $C_f + 1$ columns, so the dimensions can differ for different variables. Because the dimensions can differ for different features, so we have to introduce the unification function U.

B.2. Unification Function

In Equation (3) we define the unification function as

$$U(\boldsymbol{W}_f) = \left[\boldsymbol{w}_{\text{bias}}^{\top},\ \text{Combine}\left(\{\boldsymbol{w}_i\}_{i=1}^{C_f} \right)^{\top} \right]^{\top}.$$

Here we more formally define Combine and introduce abbreviations used in the ablation tables Tables 17, 18, 19, 20 and 21.

Specifically, for numerical features Combine returns the weight unchanged, while for categorical features it performs one of the following operations:

- ignore (*bias*) returning $\emptyset$;

- substitute with zeros (*bias_ext_catzero*): $\mathbf{0}_{C_f}$;
- sum (*bias_sum_linear*): $\sum_{i=1}^{C_f} \boldsymbol{w}_i$;
- average (*bias_avg_linear*): $\frac{1}{C_f}\sum_{i=1}^{C_f} \boldsymbol{w}_i$.

B.3. Regularization

Denote $\boldsymbol{u}_i = U(\boldsymbol{W}_i)$. The intracluster loss introduced earlier is then

$$\mathcal{L}_k^{\text{intra}} = \sum_{i \in G_k} \|\boldsymbol{u}_i - \boldsymbol{\mu}_k\| = \sum_{i=1}^{F} \|\boldsymbol{M}_{k,i}\boldsymbol{u}_i - \boldsymbol{\mu}_k\| .$$

This form is easy to compute in a vectorized fashion, so we use it to calculate the loss in practice. We call this version *hard intracluster loss*.

We introduce another, more general loss for cluster compactness that we call a *soft intracluster loss*. Now, instead of calculating the average over the distances between each cluster's centroid and *its* features, we calculate the average over the weighed distances between cluster centroids and *all* features. This is achieved by applying a softmax to the real-valued membership matrix $\boldsymbol{M}$. For hard clustering, $\boldsymbol{M}$ contains raw distances between each centroid and feature. For soft clustering, $\boldsymbol{M}$ is just the usual real-valued membership matrix. The loss is then

$$\mathcal{L}_k^{\text{intra}} = \sum_{i=1}^{F} \|\text{Softmax}\left[\boldsymbol{M}_{k,i}\right]\boldsymbol{u}_i - \boldsymbol{\mu}_k\| .$$

We consider both regularization variants for all algorithms, with the choice treated as a hyperparameter.

B.4. EMA for GMM

The expected moving average (EMA) allows us to stabilize the training by making the centroid updates more conservative. For GMM, a similar operation would involve computing an "average" between two Gaussian distributions that should also be Gaussian. This can be done using several approaches.

First, the resulting distribution can be viewed as a mixture of experts (Runnalls, 2007; Alspach and Sorenson, 2003). However, a linear mixture of two Gaussian distributions is not necessarily Gaussian. Instead, a new Gaussian distribution can be built to match the moments of the mixture distribution. This Gaussian distribution also happens to have a minimal KL distance to the original mixture distribution among the Gaussian distributions. Its mean and covariance are

$$\begin{aligned}\boldsymbol{\mu}_k &= \alpha\boldsymbol{\mu}_k + (1-\alpha)\boldsymbol{\mu}'_k,\\ \boldsymbol{\Sigma}_k &= \alpha\boldsymbol{\Sigma}_k + (1-\alpha)\boldsymbol{\Sigma}'_k + \\ &\quad + \alpha(1-\alpha)(\boldsymbol{\mu}_k - \boldsymbol{\mu}'_k)\times(\boldsymbol{\mu}_k - \boldsymbol{\mu}'_k)^T.\end{aligned}$$

Second, the resulting distribution can be viewed as a (generalized) product of experts (Hinton, 2002; Cao and Fleet, 2014). In this case, the resulting distribution is Gaussian with the following parameters

$$\begin{aligned}\boldsymbol{\Sigma}_{new}^{-1} &= \alpha\boldsymbol{\Sigma}_1^{-1} + (1-\alpha)\boldsymbol{\Sigma}_2^{-1},\\ \boldsymbol{\mu}_{new} &= \boldsymbol{\Sigma}_{new}\left(\alpha\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + (1-\alpha)\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2\right).\end{aligned}$$

Third, from the perspective of optimal transport theory, the resulting distribution can be viewed as the Wasserstein barycenter of the original Gaussian distributions (Agueh and Carlier, 2011; Cuturi and Doucet, 2014). In the case of Wasserstein-2 distance, it is also a Gaussian with mean and covariance

$$\begin{aligned}\boldsymbol{\mu}_{new} &= \alpha\boldsymbol{\mu}_1 + (1-\alpha)\boldsymbol{\mu}_2,\\ \boldsymbol{\Sigma}_{new} &= \left(\alpha\boldsymbol{\Sigma}_1^{1/2} + (1-\alpha)\boldsymbol{\Sigma}_2^{1/2}\right)^2.\end{aligned}$$

For our application, we choose to use moment matching because it is the fastest and easiest to compute. Future works can investigate whether other approaches lead to better stabilization of the GMM components during training.

B.5. Soft Membership Functions

The soft membership function transforms a real-valued characterization of the cluster membership (e.g., distances to the centroids for K-means) to a binary one. Ideally, it should produce clusters that are (a) not empty and (b) not overly full. Inability to satisfy these conditions would likely lead to non-informative feature groups and worse downstream performance. Furthermore, the function should expose a parameter to control "softness" of the learned clusters. Designing such a function for use during training is not trivial, so we considered multiple options.

First, we tried simple thresholding with the threshold parameter Δ controlling the softness.

$$\boldsymbol{M}_{k,f} = \mathbb{1}\left(\boldsymbol{p}_f^k > \Delta\right).$$

However, in this case, since the absolute values of the cluster membership (e.g. distances to centroids) are

not controlled, a good value of the threshold parameter would differ across clusters and training runs.

One way to address this challenge is to select feature groups based on percentiles instead of thresholds (i.e., choose the top 5 % features for each cluster). However, in this case the sizes of the clusters end up having a fixed size, which constrains the ability of the model to learn divers groups.

Another way to address this is to softmax the membership values before applying the threshold

$$\boldsymbol{M}_{k,f} = \mathbb{1}\left(\text{Softmax}\,\boldsymbol{p}_f^l > \Delta\right).$$

In practice, we found it very hard to tune with $\Delta \gtrsim 0.4$ producing empty clusters and $\Delta \approx 0.3$ producing full clusters. This is probably due to the distances being similar to each other in the early stages of training.

A different way to control the magnitude of the membership values is to normalize by the maximum as Equation (4). We found this approach to be easier to tune but still suffer from producing overly complete clusters. An additional practical modification that we apply is to first assign every feature to a cluster using an argmax like in hard clustering, then assign every cluster with a feature, and finally apply the soft membership function. These additional steps ensure that every feature is assigned and that every cluster is not empty, resulting in improved stability and easier hyperparameter search.

B.6. Cluster Initialization

One of the important design questions for clustering algorithms is their initialization. We consider initialization from k-means++ and expert-defined feature groups.

k-means++ is a classic algorithm used to select centroids for K-means (Arthur and Vassilvitskii, 2006). First, one feature is chosen uniformly at random as a cluster centroid. Then, the next centroid is chosen randomly among the other features with a probability proportional to the inverse square distance to the closest centroid. This process is continued until all centroids are chosen. The algorithm was designed for K-means and is therefore directly applicable in our case as well. For GMM, first we use k-means++ to initialize centroids, then we choose the closest centroid to each feature to define the initial groups, and finally we calculate the Gaussian means and covariance using these groups.

Expert-defined feature groups for ICU data were proposed by Kuznetsova et al. (2023). In this case, the expert group features are given directly as input to the model, so we calculate the K-means centroids or GMM means and covariances directly using these splits. Using expert-defined feature groups already improves downstream performance, so initializing from them can help discover even better learned feature groups.

B.7. Adaptive Number of Clusters

As discussed in Section 4 , the number of clusters that the proposed algorithm uses is technically fixed, which is a limitation. Note, however, that the algorithm does have some flexibility: it can use fewer clusters by setting some of them to be empty like in the MIMIC Decompensation case (see Figure 12).

To choose the best number of clusters we use a simple hyperparameter search. Other methods that could be used for that or for adjusting the number of clusters dynamically include stability-based method (Lange et al., 2004), the elbow method (Thorndike, 1953), and the Dirichlet processes (Ferguson, 1973).

Stability-based methods would choose the number of clusters based on the metrics that characterize the stability of clustering between bootstrapped versions of the dataset. This does not align with our main goal of improving the downstream performance of the underlying predictive model.

The elbow method could be used with the downstream performance metrics directly. However, in practice, it would require fixing all other hyperparameters. Our ablation study shows that performance can vary based on them (see Appendix C.3.6), so fixing one set of parameters for a different number of clusters might not lead to the best performance. At the same time, performing a sufficient hyperparameter search already allows us to draw plots similar to those used by the elbow method (subfigures (j) in Tables 17, 18, 19, 20 and 21). We show results for the application of the elbow method in Appendix C.3.3.

More advanced methods like Dirichlet processes would require a big modification to the pipeline and would likely be challenging to stabilize within our dynamic clustering context, which is why we leave this to future work.

Appendix C. Experiment Details

C.1. Metrics

We evaluate feature groups according to either how good the clusters are themselves or how they influence downstream performance. For the former, we use both internal metrics (do not require knowledge of the true underlying feature groups) and external metrics (vice versa).

The true feature grouping is known only for synthetic data. In this case, we can use external metrics:

- `Adjusted rand index` (ARI) represents the number of permutations of points needed to recover the true clustering from the learned one (Hubert and Arabie, 1985). It takes values between -1 and 1, higher the better (0 corresponds to random assignment and 1 corresponds to a perfect match). More formally, if the rand index RI is the proportion of pairs of points that are correctly assigned, and $\mathbb{E}(\text{RI})$ is expected to be RI under a random permutation, then

$$\text{ARI} = \frac{\text{RI} - \mathbb{E}(\text{RI})}{\max(\text{RI}) - \mathbb{E}(\text{RI})}.$$

- `Normalized mutual information` (NMI) is derived from information theory and represents the mutual information between true and predicted clusterings (Vinh et al., 2009). It takes values between zero and one; the higher, the better. Denoting I mutual information, and H as entropy,

$$\text{NMI}(u, v) = \frac{I(u, v)}{\sqrt{H(u)H(v)}}.$$

Without access to the true feature grouping, the quality of the clusters can be evaluated based on the distances between the clusters and points (internal metrics). For example, we use intercluster and intracluster distances for regularization Equation (5). Additionally, for synthetic data, we compute the

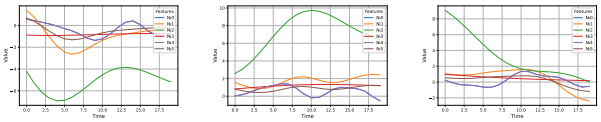
- `Silhoette score` (Rousseeuw, 1987). Denoting mean distances between a feature f to the points in the same cluster $\mathcal{D}_{\text{same}}$ and to the points in the nearest cluster $\mathcal{D}_{\text{nearest}}$

$$S = \mathbb{E}_i \left[\frac{\mathcal{D}_{\text{nearest}} - \mathcal{D}_{\text{same}}}{\max\left(\mathcal{D}_{\text{nearest}}, \mathcal{D}_{\text{same}}\right)} \right].$$

For downstream performance, we consider metrics that are commonly used in the field and suit highly imbalanced tasks on ICU data (Yèche et al., 2021; Harutyunyan et al., 2019). Specifically, for classification, we use `area under the precision-recall curve` (AUPRC) and `area under the receiver-operator curve` (AUROC).

C.2. Synthetic Data

We generate synthetic data by sampling 10,000 examples from Gaussian processes with 6 features and the length of 20. We use the RBF kernel for all features with length scales $\{1, 2, 4, 8, 1, 2\}$ and amplitudes $\{0.5, 1.0, 3.5, 0.5, 0.5, 0.5\}$.



(a) Sample 0 (b) Sample 1 (c) Sample 2

Figure 5: Samples from the synthetic dataset.

The data transformations for static clustering are formally defined as

$$\mathcal{D}_{\text{flat}} = \left\{ \Big\|_{i,t} \left(\boldsymbol{x}^i_{t,f} \mid y^i \right) \right\}_{f=1}^{F},$$

$$\mathcal{D}_{\text{time mean}} = \left\{ \Big\|_{i} \left(\frac{1}{D} \sum_t \boldsymbol{x}^i_{t,f} \mid y^i \right) \right\}_{f=1}^{F},$$

$$\mathcal{D}_{\text{sample mean}} = \left\{ \Big\|_{t} \left(\frac{1}{T} \sum_i \boldsymbol{x}^i_{t,f} \mid y^i \right) \right\}_{f=1}^{F},$$

$$\mathcal{D}_{\text{full mean}} = \left\{ \frac{1}{DT} \sum_{i,t} \boldsymbol{x}^i_{t,f} \mid y^i \right\}_{f=1}^{F}.$$

Here, $\|$ and $\mid$ are concatenation operations.

We train the model for 1000 epochs with a learning rate 0.002, an early stopping patience of 10 and a batch size of 5000. Clustering is done using Hard K-means with k-means++ initialization and no regularization. FT-Transformer has 6 hidden units, 2 heads, and a depth of 1. Further details can be found in the source code.

Table 3: **Performance benchmark for different grouping schemes** (AUROC ↑) on online and offline classification tasks using ICU datasets. Mean and standard deviation are reported over three training runs with different random seeds. Best results are highlighted in bold. Learned groupings perform comparably to expert-defined ones. Reference results for prior groups (Prior) and models without feature grouping (None) are taken from Kuznetsova et al. (2023).

Feature Grouping			HIRID		MIMIC-III	
Type	Membership	Algorithm	Circ	Mort	Decomp	Mort
None	–	–	0.911 ± 0.001	0.905 ± 0.002	**0.916** ± 0.001	0.858 ± 0.002
Prior	Hard	Type	0.910 ± 0.002	0.901 ± 0.003	0.914 ± 0.001	0.860 ± 0.002
Prior	Hard	Organ	**0.916** ± 0.0003	**0.910** ± 0.003	0.914 ± 0.001	**0.861** ± 0.002
Learned	Hard	K-means	0.912 ± 0.0008	**0.912** ± 0.004	**0.917** ± 0.0002	**0.863** ± 0.002
Learned	Hard	Fuzzy K-means	0.907 ± 0.0004	0.908 ± 0.003	**0.916** ± 0.0009	**0.861** ± 0.004
Learned	Hard	GMM	0.912 ± 0.0008	0.900 ± 0.004	**0.917** ± 0.0009	0.858 ± 0.001
Learned	Soft	Fuzzy K-means	**0.911** ± 0.0022	0.902 ± 0.002	**0.916** ± 0.0019	**0.863** ± 0.002
Learned	Soft	GMM	**0.912** ± 0.0012	0.906 ± 0.001	**0.915** ± 0.0024	**0.861** ± 0.003

C.3. Medical Data

C.3.1. Task Definitions

We consider clinically relevant tasks defined for the HiRID and MIMIC-III datasets (Yèche et al., 2021; Harutyunyan et al., 2019). These can be split into static (prediction is made once per sample) and dynamic tasks (prediction is made continuously for the sample); and into classification and regression tasks.

On HiRID we consider

- `Circulatory failure` (Circ), a dynamic binary classification task predicting whether a circulatory failure will occur in the next 8 hours;
- `Mortality` (Mort), a static binary classification predicting whether the patient will pass away at the end of the admission using observation from the first 24 hours;

On MIMIC-III we consider

- `Decompensation` (Decomp), a dynamic binary classification task predicting whether a patient will pass away in the next 24 hours;
- `Mortality` (Mort), a static binary classification predicting whether the patient will pass away at the end of the admission using observation from the first 24 hours.

More detailed definitions can be found in Kuznetsova et al. (2023); Yèche et al. (2021); Harutyunyan et al. (2019).

C.3.2. Setup

We train using 1 to 4 NVidia RTX2080TI GPUs with Xeon E5-2630v4 CPUs. The batch size ranges from 1 to 16 depending on the model size and hardware capacity. The training time is between 5–22 hours. We use Adam optimizer and early stopping with patience of 10 epochs (Diederik, 2014). The models are implemented in PyTorch (Paszke et al., 2019) and scikit-learn (Pedregosa et al., 2011).

We perform a hyperparameter search across the number of clusters, type of embedding extraction function U, EMA coefficients, regularization type and coefficient, cluster initialization type, fuzzy coefficient (for fuzzy K-means), covariance matrix type (for GMM), membership coefficient (for soft membership models). The parameters for the underlying embedding and sequence model are taken from Kuznetsova et al. (2023); Yèche et al. (2021). Further details can be found in the project repository.

For t-SNE visualizations, we use a perplexity of 10 for HiRID and 5 for MIMIC with a random seed of 42 (van der Maaten and Hinton, 2008).

C.3.3. Additional Results

Additional metrics Similarly to Table 2, we present the downstream performance now measured by AUROC in Table 3. Our conclusions are consistent for both metrics.

We provide p-values for the paired Student t-test (Student, 1908) on the null hypothesis that a model with grouping performs better than the reference model without grouping in terms of AUPRC in Table 4.

Table 4: P-values for comparison of various models to the reference model without feature grouping by AUPRC. Values under 0.05 are highlighted in bold.

Grouping	HiRID Circ	HiRID Mort	MIMIC Decomp	MIMIC Mort
Prior by Type	**0.0348**	0.1745	0.0774	0.1890
Prior by Organ	**0.0167**	0.1045	**0.0109**	0.0774
Learned Hard K-means	0.3651	**0.0187**	0.0530	0.0927
Learned Hard Fuzzy K-means	0.4335	0.0548	0.8361	0.1275
Learned Hard GMM	0.0827	**0.0151**	0.0620	0.4275
Learned Soft Fuzzy K-means	0.1719	0.0982	0.0833	0.2660
Learned Soft GMM	**0.0286**	0.7364	**0.0460**	0.2433

In addition, we provide an F-score (Goutte and Gaussier, 2005) and the Matthews correlation coefficient (MCC) (Chicco and Jurman, 2023) for the mortality prediction experiments. These metrics were proposed to address limitations of AUROC and AUPRC in evaluating performance on imbalanced tasks, but are not yet widely used in ICU research (Chicco and Jurman, 2023; Yadav and Subbian, 2025).

Table 5: MCC and F1 scores for various clustering algorithms trained for mortality prediction.

Algorithm	MCC		F1	
	MIMIC Mort	HiRID Mort	MIMIC Mort	HiRID Mort
K-means	0.4467 ± 0.0097	0.5838 ± 0.0030	0.5142 ± 0.0090	0.6171 ± 0.0012
Fuzzy K-means	0.4538 ± 0.0083	0.5837 ± 0.0071	0.5113 ± 0.0023	0.6139 ± 0.0031
GMM	0.4458 ± 0.0070	0.5804 ± 0.0069	0.5046 ± 0.0074	0.6118 ± 0.0037
Soft Fuzzy GMM	0.4453 ± 0.0041	0.5661 ± 0.0065	0.5104 ± 0.0072	0.5976 ± 0.0050
Soft GMM	0.4459 ± 0.0106	0.5797 ± 0.0074	0.5110 ± 0.0075	0.6104 ± 0.0102

Learning dynamics We investigated the cluster learning dynamics and describe them for various models using a fixed seed in Figures 9, 10, 11, 12, 13 and 14. Generally, we observe that in several cases the models keep the groupings constant from initialization. We believe this to be the case of the exploration–exploitation trade-off: the earlier the clustering is set, the more time and signal passes through the other parts of the architecture, potentially improving the downstream performance. To control the trade-off, we can adjust the regularization and EMA coefficients. It also seems that the Fuzzy K-means model is particularly suitable for exploration.

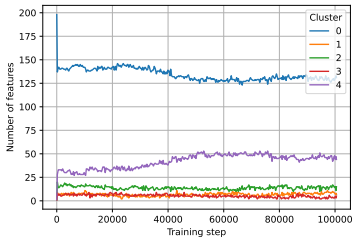
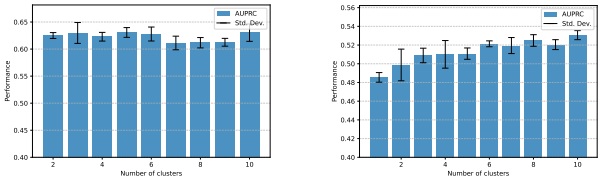


Figure 6: Dynamics of cluster sizes for the Fuzzy K-means trained for circulatory prediction on HiRID.

Impact of initialization As we discussed in Section B.6, our method can use the prior feature grouping as the initialization, as well as initialize from the k-means++ algorithm. From the ablation studies, we see that k-means++ performs better for Soft GMM trained on HiRID circulatory failure, K-means trained on HiRID mortality, and K-means trained on MIMIC mortality Tables 17(a), 18(a) and 21(a). Priors perform better for Fuzzy K-means trained on HiRID mortality and GMM trained on decompensation Tables 19(a) and 20(a). This demonstrates that our approach does not depend entirely on a good initialization from priors, but rather can leverage them to achieve better performance if algorithmic initialization does not work well.

Elbow method Here we investigate whether the Elbow method can be effectively used for the selection of the optimal number of clusters. We fix the hyperameters used by the best performing models on HiRID and MIMIC mortality prediction and vary the number of clusters. Resulting performance is shown in Figure 7. We see that the difference in performance is comparable to the that achieved by altering other hyperparameters Tables 18 and 21. This suggests that the original approach of searching for the best joint set of hyperparameters is more beneficial for the downstream performance.

Attention weights Our method includes an additional explainability mechanism based on attention weights, which is active when attention aggregation is used. This is the case for the best Fuzzy K-means trained on the HiRID dataset for circulatory failure



(a) HiRID (b) MIMIC

Figure 7: Elbow method for optimal number of clusters for mortality prediction.

prediction (other models for this task use mean aggregation, see Table 2). We visualize these weights over groups and the time before the events for three randomly selected samples in Figure 8. We see that the model consistently ranks the clusters: cluster 4 has the highest weight, followed by clusters 3, 4, and 0, while cluster 1 gets the lowest. The size of the clusters suggests that this ranking does not simply give more weight to the larger clusters (see Figure 6). We analyze the feature assignments shown in Table 10: cluster 1 seems to combine variables on acute neurological event and resuscitation; cluster 0 is the largest and contains a mix of diverse features; cluster 2 contains a combination of variables related to sepsis and acute pancreatic conditions; cluster 3 might indicate acute respiratory distress (ARDR) if the underlying ventilator support (AWPmean) and deep sedation (deep sedation) variables are high, it also curiously seemingly unrelated coagulation variable (FVII); cluster 4 contains many dynamic variables that might indicate patient crashing (key vitals: RR, HR; key labs: a_Lac, CRP). Attention therefore places more weight on clusters that indicate patient crash and respiratory distress, while placing less weight on a large group of mixed variables and a group specifically targeting acute neurological conditions. Although insightful, this interpretation should be viewed with caution because the model underperforms other clustering algorithms in terms of the downstream metric.

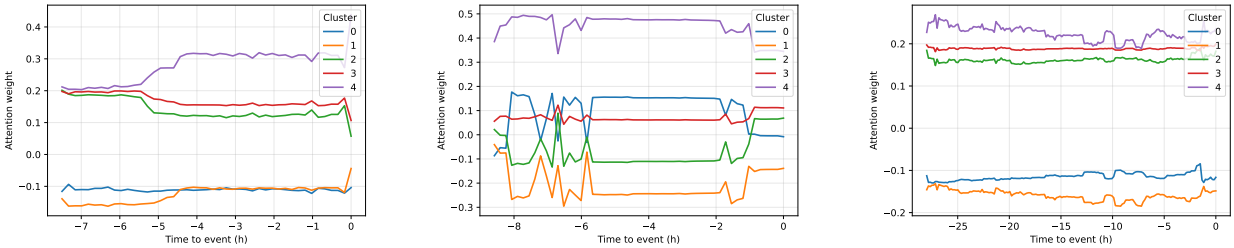
C.3.4. Learned Groups

The feature groups learned on MIMIC for decompensation and mortality prediction are shown in Tables 6, 7 and 8. The feature groups learned on HiRID for circulatory failure and mortality prediction in Tables 9, 11 and 12. We present results for the best model if one stands out clearly, and for the top two models if multiple models are competitive (see Table 2). We compare (1) the learned groups with the expert-defined groups, (2) the groups learned on the same dataset but for different tasks, and (3) the groups learned by different models on the same dataset and the same task.

MIMIC decompensation Fuzzy K-means shows the best downstream performance for this task. As discussed previously (see Section 3.2), from the learning dynamics (see Figure 12) we see that the model initially splits the features into four groups and eventually learns to assign all the features to one cluster (see Table 6). This replicates the best prior grouping that also just assigns all features to a single group (see Table 2).

MIMIC mortality The best two models for this task are K-means and fuzzy K-means. Both models produce eight clusters, with four or five of them containing a single feature (see Tables 7 and 8)). The K-means appears to select groups that focus on identifying particular conditions such as shock (cluster 0), inflammation (cluster 1), acidoses (cluster 2), etc. or baseline state of the patient such as their constitution (cluster 7) and organ function (clusters 1 and 6) Table 7). The Fuzzy K-means produces clusters of similar sizes. Some of them are quite similar to what K-means produces: cluster 1 is exactly the same (respiratory rate), cluster 5 (systolic blood pressure) and cluster 6 (mean blood pressure) describe blood pressure, clusters 6 (weight) and cluster 3 (height) describe patient's size. Cluster 5 could be related to septic shock: low diastolic blood pressure and high glucose could be its indicators. Other clusters seem more mixed: cluster 0 contains variables that describe vital signs and neurological status, cluster 8 describes gas exchange and hemodynamics. These clusters (and the dataset in general) do not contain many variables, so we do not see an obvious pattern here.

Surprisingly, the models divide some naturally related variables, such as respiratory and hemodynamic features, into different groups (e.g., K-means assigns oxygen saturation and respiratory rate to clusters 0 and 5; blood pressure and heart rate to clusters 0 and 7). It is not immediately clear what the benefit is, but perhaps splitting the variables allows capturing the relation between different organ systems within one cluster for multiple clusters, enabling better characterization of certain conditions. Specifically, the model might need the features characterizing the res-



(*a*) Sample 90 (*b*) Sample 120 (*c*) Sample 336

Figure 8: Attention weights between clusters and over time before the event produced by Fuzzy K-means trained for circulatory failure prediction on HiRID.

piratory system in both groups that describe coma and hypoxia.

MIMIC overall We notice that the models select only one cluster for decompensation and eight for both models for mortality. We speculate that this difference in behavior occurs from the nature of the tasks. Decompensation is an online task, where the model should capture the rapid deterioration of the patient's condition. Mortality is an offline prediction task, where the model is supposed to assess the patient's general status over the first two days and relate it to the outcome after the whole, possibly multiday ICU stay. This means that models trained for decompensation prediction might not rely on the predictive groups, but rather on how quickly all features change; whereas models trained for mortality prediction would rely on identifying whether the patient has a particular condition (e.g., shock) or if they are likely to develop it based on their overall status (e.g., weight, respiratory support, etc.).

HiRID circulatory failure The soft GMM model shows the best downstream results for this task. In Table 9 we see three clusters that single out three laboratory measurements: creatine kinase (cluster 1), alkaline phosphatase (cluster 2), and creatine kinase-MB (cluster 3). These could be indicators of specific critical events: a spike in creatine kinase could suggest rhabdomyolysis, creatine kinase-MB could indicate biliary sepsis, and alkaline phosphatase could point to acute myocardial infarction. However, other variables could also indicate these conditions, so it is not completely certain that the model uses these variables to predict these specific conditions. The other two clusters are quite large, containing the rest of over 200 variables from HiRID. Interpreting them also becomes harder, with both containing vitals, laboratory analysis, and treatments for various organ systems at the same time.

HiRID mortality For mortality, the best downstream results are produced by K-means and Fuzzy K-means models. Both produce a similar number of clusters (4 and 5 respectively) that have somewhat similar sizes (except for cluster 2 of Fuzzy K-means) (see Tables 11 and 12). Here, as in the circulatory failure case, interpreting the clusters is not straightforward. Cluster 0 from K-means, for example, contains many hemodynamic variables (e.g., WBC, ferritin, Ca, etc.) but at the same time contains electrolytes (e.g., Na, K); cluster 1 contains numerous vital signs (e.g., HR, SpO2), but at the same time contains treatments (e.g., antiviral, immunosuppression, milrinone). Similarly, for Fuzzy K-means, cluster 1 contains coagulation factors (FII, FV, FVII, FX) but at the same time cardiologic variables (e.g., ABP, ZVD), treatments (e.g., antibiotics, cortisol) and other variables. The only small feature cluster here, cluster 2, could be related to cardiopulmonary management during resuscitation.

HiRID overall Across tasks, we find that models use a similar number of clusters for HiRID. Due to the large number of variables, they become difficult to access. Due to computational cost, we considered only up to 8, which seems too few to the over 200 features that HiRID has.

Consistent with MIMIC, we find that the smaller clusters seem to track specific acute conditions. This finding could help develop new feature groups tailored to specific tasks or patient cohorts.

C.3.5. Prior Feature Groups

The prior grouping by organ system for HiRID and MIMIC-III is presented in Tables 13 and 14, respectively. Prior grouping by measurement type is shown in Tables 15 and 16. The tables are reproduced with minor modifications from Kuznetsova et al. (2023). For a description of individual variables, see Hyland et al. (2020) and Johnson et al. (2016).

C.3.6. Ablation Studies

Ablation studies are shown in Tables 17, 18, 19, 20 and 21. We discuss them in Section 3.2.

Generally, we find that a number of clusters of 3–5, a small EMA rate of 0.25, and a regularization weight of 0.0001 generally leads to good performance and can be used as a good first set of parameters. However, the choice of the initialization, architecture functions, and regularizer type depends on the underlying dataset and task, and should be optimized. To balance flexibility and simplicity, we suggest starting with k-means++, bias unification, attention merging, and a hard regularizer.

The training time of the best-performing runs for circulatory failure prediction on HiRID was on average 348 minutes without grouping, 403 minutes with prior organ grouping, 368 minutes for K-means, 299 minutes for Fuzzy K-means, 544 minutes for Soft Fuzzy K-means, 305 minutes for GMM, and 303 minutes for Soft GMM. For other tasks the training times of different models were similarly close, averaging to around 96 minutes for mortality prediction on HiRID, 359 for decompensation prediction on MIMIC, and 95 for mortality prediction on MIMIC (note that the number of GPUs and the batch size varies between tasks).

Table 6: Feature grouping learned by K-means model of MIMIC features for decompensation prediction. Interpreted as no grouping is necessary.

Group name	Features
Cluster 0: all features	Capillary refill rate, Glasgow coma scale eye opening, Glasgow coma scale motor response, Glasgow coma scale total, Glasgow coma scale verbal response, Diastolic blood pressure, Fraction inspired oxygen, Glucose, Heart Rate, Mean blood pressure, Oxygen saturation, Respiratory rate, Systolic blood pressure, Temperature, pH

Table 7: Feature grouping learned by K-means model of MIMIC features for mortality prediction. Interpreted as grouping by baseline status and monitoring of certain conditions.

Group name	Features
Cluster 0: "coma status"	Glasgow coma scale eye opening, Glasgow coma scale verbal response, Time, Mean blood pressure, Oxygen saturation
Cluster 1: "respiratory status"	Respiratory rate
Cluster 2: "inflammation status"	Temperature
Cluster 3: "acidosis status"	pH
Cluster 4: "hypoxia status"	Glasgow coma scale motor response, Height, Fraction inspired oxygen, Glucose
Cluster 5: "systolic function status"	Systolic blood pressure
Cluster 6: "baseline constitution"	Weight
Cluster 7: "hemodynamic stability"	Capillary refill rate, Glasgow coma scale total, Diastolic blood pressure, Heart Rate

Table 8: Feature grouping learned by Fuzzy K-means model of MIMIC features for mortality prediction. Interpreted as grouping by baseline status and monitoring of certain conditions.

Group name	Features
Cluster 0: "mixed vitals and neurological status"	Capillary refill rate, Glasgow coma scale eye opening, Glasgow coma scale total, Glasgow coma scale verbal response, Time, Heart Rate, Temperature
Cluster 1: "respiratory status"	Respiratory rate
Cluster 3: "baseline statury"	Height
Cluster 4: "motor status"	Glasgow coma scale motor response
Cluster 5: "septic shock"	Diastolic blood pressure, Glucose
Cluster 6: "circulatory status"	Mean blood pressure
Cluster 7: "respiratory support"	Fraction inspired oxygen
Cluster 8: "mixed gas exchange and hemodynamics"	Oxygen saturation, Systolic blood pressure, Weight, pH

Table 9: Feature grouping learned by Soft GMM model of HiRID features for circulatory failure prediction. Interpreted as isolated specific indicators and two large mixed feature groups.

Group name	Features
Cluster 0: "mixture class"	OUTurine/h, Administriation of antibiotics, administration of antiviral, Acetazolamide, non-steroids, Parenteral Feeding, a-BE, Ca, Ketalar, Antiepileptica, Anti delirant medi, aPTT, Fibrinogen, Muskelrelaxans, Anexate, Cl-, Factor VII, Neutr, Amiodaron, TSH, T Central, ABPs, ABPd, GCS Antwort, Glucose Administration, PAPd, PCWP, Steroids, Others in Case of HIT, Marcoumar, Propofol, Protamin, Anti Fibrinolyticum, Kalium, Non-opioide, a_Hb, a_Lac, Incrys, Incolloid, packed red blood cells, platelets, coagulation factors, pH Liquor, Plateaudruck, theophyllin, C-reactive protein, AWPmean, RR set, vasodilatators, ACE Inhibitors, Adenosin, Ca2+ total, Digoxin, a_pCO2, INR, albumin, phosphate, Ammoniak, Hb, a_PO2, Mg_lab, platelet count, Atropin, Urea, Mineralokortikoid, Segm. Neut., Terlipressin, BNP, urinary creatinin, urinary Na+, urinary urea, Lipase, AMYL-S
Cluster 1: "rhabdomyolysis indicator"	creatine kinase
Cluster 2: "biliary sepsis indicator"	alkaline phosphatase
Cluster 3: "acute myocardial infarction indicator"	creatine kinase-MB
Cluster 4: "mixture class"	NIBPs, Administation of antimycotic, RR, K-sparend, NIBPd, NIBPm, GCS Motorik, Insuling Langwirksam, GCS Augenoffnen, PAPm, PAPs, ICP, supplemental oxygen, Aldosteron Antagonist, Enteral Feeding, ZVD, Chemotherapie, Immunoglobulin, Immunsuppression, Phosphat, Na, a_MetHb, OUT, NSAR, v-Lac, a_HCO3-, Peripherial Anesthesia, K+, Na+, FII, Parkinson Medikaiton, dobutamine, Sartane, Factor V, Ca Antagonists, B-Blocker, Andere, Ca2+ ionizied, Beh. Pulm. Hypertonie, Stabk. Neut., Cortisol, HR, ASAT, SpO2, ABPm, ALAT, ETCO2, RASS, Antihelmenticum, TOF, bilirubine; total, CO, SvO2(m), Loop diuretics, Thiazide, Haemofiltration, steroids, FIO2, ST1, Insulin Kurzwirksam, Peep, Thrombozytenhemmer, Heparin, NMH, Benzodiacepine, Alpha 2 Agonisten, Barbiturate, ST2, Liquor/h, ST3, Lysetherapie, Rhythmus, IN, Pankreas Enzyme, VitB Substitution, Nimodipin, Opiate, Weight, Mg, a_COHb, Trace elements, Bicarbonate, GCSF, FFP, a_pH, Psychopharma, norepinephrine, epinephrine, Ventilator mode, TV, Naloxon, Spitzendruck, milrinone, levosimendan, vasopressin, procalcitonin, lymphocyte, desmopressin, Laktat Liquor, AiwayCode, factor X, glucose, total white blood cell count, a_SO2, Zentral venöse sättigung, Bilirubin; direct, Thyrotardin, Thyroxin, Thyreostatikum, MCH, gamma-GT, MCHC, Antihistaminka, Troponin-T, pH Drain, Glucose Liquor, creatinine, MCV, BSR, AMYL-Drainag, Ferritin

Table 10: Feature grouping learned by Fuzzy K-means model of HiRID features for circulatory failure prediction.

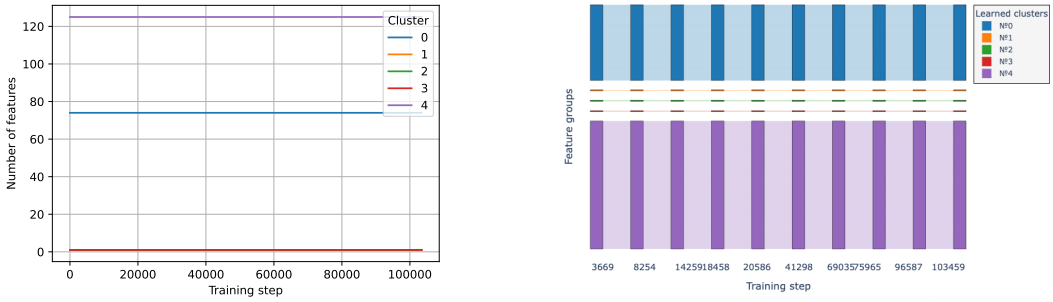
Group name	Features
Cluster 0: "mixed"	OUTurine/h, NIBPs, Administation of antimycotic, K-sparend, NIBPd, GCS Motorik, Insuling Langwirksam, GCS Augenoffnen, administration of antiviral, PAPm, PAPs, ICP, supplemental oxygen, Aldosteron Antagonist, Enteral Feeding, ZVD, non-steroids, Chemotherapie, Parenteral Feeding, Immunoglobulin, Immunsuppression, Phosphat, Na, a-BE, a_MetHb, Ca, OUT, NSAR, v-Lac, Ketalar, a_HCO3-, Peripherial Anesthesia, Antiepileptica, Anti delirant medi, Fibrinogen, K+, Na+, Parkinson Medikaiton, dobutamine, Sartane, Cl-, Factor V, Ca Antagonists, B-Blocker, Andere, Beh. Pulm. Hypertonie, Neutr, Stabk. Neut., None, Cortisol, None, T Central, ABPd, ASAT, ABPm, Glucose Administration, RASS, PAPd, PCWP, Antihelmenticum, TOF, bilirubine; total, Steroids, CO, SvO2(m), Loop diuretics, Thiazide, Haemofiltration, steroids, FIO2, Thrombozytenhemmer, NMH, Others in Case of HIT, Benzodiacepine, Alpha 2 Agonisten, Barbiturate, Marcoumar, ST2, Protamin, Anti Fibrinolyticum, Liquor/h, Rhythmus, Pankreas Enzyme, VitB Substitution, Kalium, Non-opioide, Mg, a_Hb, Trace elements, Incolloid, packed red blood cells, Bicarbonate, FFP, a_pH, norepinephrine, epinephrine, TV, Naloxon, Spitzendruck, Plateaudruck, milrinone, levosimendan, theophyllin, vasopressin, procalcitonin, desmopressin, vasodilatators, Aiway-Code, Ca2+ total, albumin, glucose, total white blood cell count, a_SO2, Mg_lab, None, alkaline phosphatase, Atropin, Thyrotardin, Urea, Mineralokortikoid, MCH, MCHC, Segm. Neut., Terlipressin, Troponin-T, creatinine, AMYL-Drainag, urinary Na+, AMYL-S
Cluster 1: "acute neurological status"	ETCO2, Lysetherapie, Nimodipin, Incrys, coagulation factors, pH Liquor, Laktat Liquor, a_pCO2, BSR, BNP
Cluster 2: "sepsis and pancreatic dysfunction"	Administriation of antibiotics, ALAT, Heparin, Propofol, ST3, Weight, platelets, lymphocyte, phosphate, Thyroxin, Thyreostatikum, Antihistaminka, Lipase
Cluster 3: "sever ARDS + FVII"	Factor VII, Opiate, AWPmean
Cluster 4: "dynamic shock and inflammation"	RR, NIBPm, Acetazolamide, aPTT, Muskelrelaxans, Anexate, FII, Ca2+ ionizied, Amiodaron, TSH, HR, ABPs, SpO2, GCS Antwort, ST1, Insulin Kurzwirksam, Peep, IN, a_COHb, a_Lac, GCSF, Psychopharma, Ventilator mode, C-reactive protein, RR set, ACE Inhibitors, Adenosin, factor X, Digoxin, INR, Ammoniak, Hb, a_PO2, Zentral venöse sättigung, None, platelet count, Bilirubin; direct, gamma-GT, None, pH Drain, Glucose Liquor, creatine kinase, MCV, creatine kinase-MB, Ferritin, urinary creatinin, urinary urea

Table 11: Feature grouping learned by K-means model of HiRID features for mortality prediction. Interpreted as mixed grouping, with clusters generally containing large subgroups of variables relating to different systems and conditions.

Group name	Features
Cluster 0: "cardiovascular, multi-organ variables"	Ca, dobutamine, Ca Antagonists, Andere, Ca2+ ionizied, Beh. Pulm. Hypertonie, Amiodaron, ABPd, PAPd, FIO2, Others in Case of HIT, Barbiturate, Marcoumar, VitB Substitution, a_Hb, epinephrine, Naloxon, procalcitonin, desmopressin, Laktat Liquor, INR, phosphate, total white blood cell count, Bilirubin; direct, MCH, pH Drain, MCV, AMYL-Drainag, Ferritin, Lipase
Cluster 1: "vitals, multi-organ variables"	RR, administration of antiviral, PAPm, ICP, Enteral Feeding, non-steroids, Parenteral Feeding, Immunsuppression, Cl-, HR, SpO2, Glucose Administration, RASS, SvO2(m), Thrombozytenhemmer, Heparin, Benzodiacepine, Anti Fibrinolyticum, Kalium, Weight, Trace elements, GCSF, a_pH, Plateaudruck, milrinone, ACE Inhibitors, Ca2+ total, a_pCO2, alkaline phosphatase, Thyrotardin, Urea, Thyroxin, Mineralokortikoid, gamma-GT, Troponin-T, BSR
Cluster 2: infection management, sedation	Administriation of antibiotics, Administation of antimycotic, NIBPd, supplemental oxygen, Chemotherapie, Phosphat, Ketalar, Anti delirant medi, aPTT, K+, B-Blocker, TSH, ABPs, PCWP, Antihelmenticum, Loop diuretics, Thiazide, Alpha 2 Agonisten, Propofol, ST2, Protamin, a_COHb, AWPmean, vasodilatators, glucose, Ammoniak, Zentral venöse sättigung, Antihistaminka, creatine kinase-MB, urinary urea, AMYL-S
Cluster 3: "fluids, inflammation"	ZVD, Immunoglobulin, a-BE, a_MetHb, OUT, a_HCO3-, Peripherial Anesthesia, Fibrinogen, Muskelrelaxans, Anexate, Na+, FII, Parkinson Medikaiton, Factor VII, Haemofiltration, steroids, IN, Incrys, Incolloid, Bicarbonate, FFP, Psychopharma, Ventilator mode, pH Liquor, Spitzendruck, theophyllin, C-reactive protein, RR set, factor X, Digoxin, albumin, Mg_lab, Atropin, Thyreostatikum, MCHC, Terlipressin, Glucose Liquor, creatinine
Cluster 4: "vasoactive drugs, shock"	OUTurine/h, NIBPs, K-sparend, NIBPm, GCS Motorik, Insuling Langwirksam, GCS Augenoffnen, PAPs, Aldosteron Antagonist, Acetazolamide, Na, NSAR, v-Lac, Antiepileptica, Sartane, Factor V, Neutr, Stabk. Neut., Cortisol, T Central, ASAT, ABPm, ALAT, ETCO2, GCS Antwort, TOF, bilirubine; total, Steroids, CO, ST1, Insulin Kurzwirksam, Peep, NMH, Liquor/h, ST3, Lysetherapie, Rhythmus, Pankreas Enzyme, Nimodipin, Opiate, Non-opioide, Mg, a_Lac, packed red blood cells, platelets, coagulation factors, norepinephrine, TV, levosimendan, vasopressin, lymphocyte, AiwayCode, Adenosin, Hb, a_PO2, a_SO2, platelet count, Segm. Neut., creatine kinase, BNP, urinary creatinin, urinary Na+

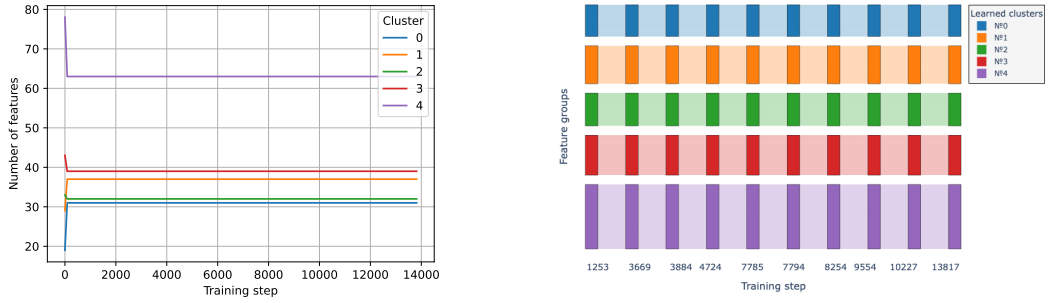
Table 12: Feature grouping learned by Fuzzy K-means model of HiRID features for mortality prediction. Interpreted as mixed grouping, with clusters generally containing large subgroups of variables relating to different systems and conditions.

Group name	Features
Cluster 0: monitoring, treatments, mixed	OUTurine/h, NIBPs, Administation of antimycotic, NIBPm, GCS Motorik, administration of antiviral, Aldosteron Antagonist, Immunsuppression, Phosphat, a_MetHb, Ca, OUT, Ketalar, a_HCO3-, Anti delirant medi, aPTT, Muskelrelaxans, FII, Parkinson Medikaiton, Cl-, Ca Antagonists, Amiodaron, GCS Antwort, Glucose Administration, PCWP, bilirubine; total, Loop diuretics, NMH, Benzodiacepine, Liquor/h, Non-opioide, Trace elements, Spitzendruck, theophyllin, RR set, Laktat Liquor, AiwayCode, Ca2+ total, a_pCO2, INR, phosphate, total white blood cell count, a_PO2, alkaline phosphatase, Atropin, Thyreostatikum, gamma-GT, Terlipressin, Troponin-T, Glucose Liquor, MCV, BSR, AMYL-Drainag, Ferritin, AMYL-S
Cluster 1: sefation, shock, mixed	K-sparend, Insuling Langwirksam, Parenteral Feeding, v-Lac, Neutr, Stabk. Neut., TSH, HR, RASS, TOF, steroids, Others in Case of HIT, Alpha 2 Agonisten, Barbiturate, Marcoumar, Propofol, VitB Substitution, Nimodipin, Opiate, a_Lac, Bicarbonate, Psychopharma, milrinone, Thyrotardin, Urea
Cluster 2: cardiopulmonary management during resuscitation	CO, Peep, Incrys
Cluster 3: hemodynamics, respiratory, infection, mixed	Administriation of antibiotics, RR, NIBPd, GCS Augenoffnen, PAPm, PAPs, ICP, supplemental oxygen, Acetazolamide, Enteral Feeding, ZVD, non-steroids, Chemotherapie, Immunoglobulin, Na, a-BE, NSAR, Peripherial Anesthesia, Antiepileptica, Fibrinogen, K+, Anexate, Na+, dobutamine, Sartane, Factor V, B-Blocker, Andere, Ca2+ ionizied, Beh. Pulm. Hypertonie, Factor VII, Cortisol, T Central, ABPs, ABPd, ASAT, SpO2, ABPm, ALAT, ETCO2, PAPd, Antihelmenticum, Steroids, SvO2(m), Thiazide, Haemofiltration, FIO2, ST1, Insulin Kurzwirksam, Thrombozytenhemmer, Heparin, ST2, Protamin, Anti Fibrinolyticum, ST3, Lysetherapie, Rhythmus, IN, Pankreas Enzyme, Kalium, Weight, Mg, a_COHb, a_Hb, Incolloid, packed red blood cells, GCSF, FFP, a_pH, platelets, coagulation factors, norepinephrine, epinephrine, Ventilator mode, TV, Naloxon, pH Liquor, Plateaudruck, levosimendan, vasopressin, C-reactive protein, AWPmean, procalcitonin, lymphocyte, desmopressin, vasodilatators, ACE Inhibitors, Adenosin, factor X, Digoxin, albumin, glucose, Ammoniak, Hb, a_SO2, Zentral venöse sättigung, Mg_lab, platelet count, Bilirubin; direct, Thyroxin, Mineralokortikoid, MCH, MCHC, Antihistaminka, Segm. Neut., pH Drain, creatine kinase, creatinine, creatine kinase-MB, BNP, urinary creatinin, urinary Na+, urinary urea, Lipase



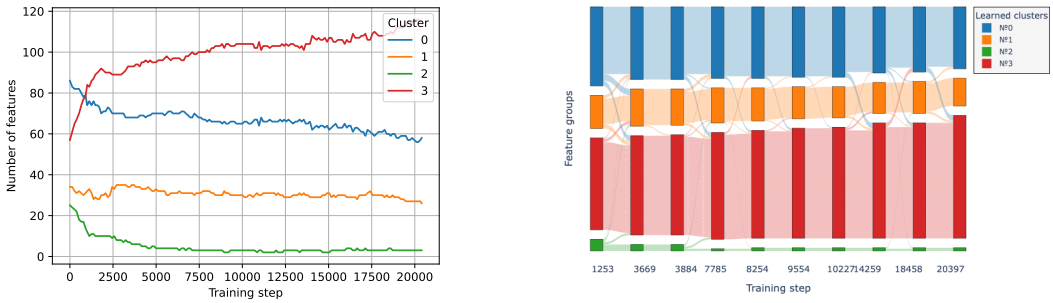
(*a*) Cluster sizes (*b*) Sankey diagram

Figure 9: Dynamics of feature assignments to clusters for Soft GMM model trained on HiRID for circulatory failure prediction. Grouping does not change from the initialization.



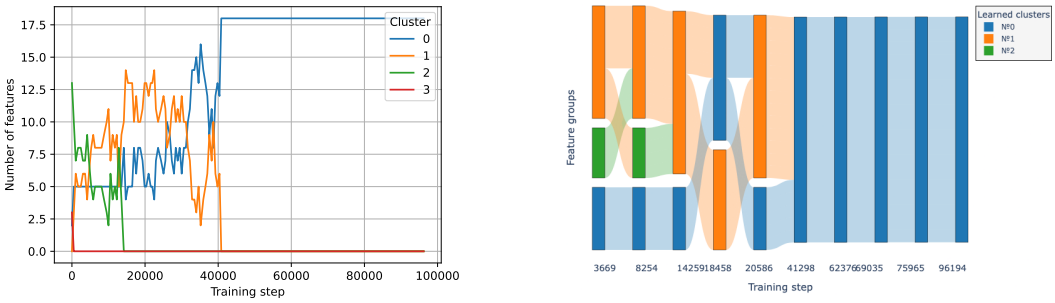
(*a*) Cluster sizes (*b*) Sankey diagram

Figure 10: Dynamics of feature assignments to clusters for K-means model trained on HiRID for mortality prediction. The groups settle in the first few iterations and stay constant throughout the rest of the training.



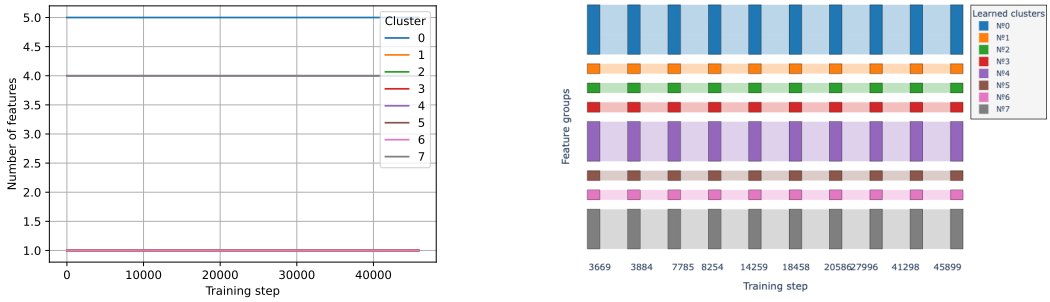
(*a*) Cluster sizes (*b*) Sankey diagram

Figure 11: Dynamics of feature assignments to clusters for Fuzzy K-means model trained on HiRID for mortality prediction. The groups slowly evolve, with the fourth cluster gradually increasing in size.



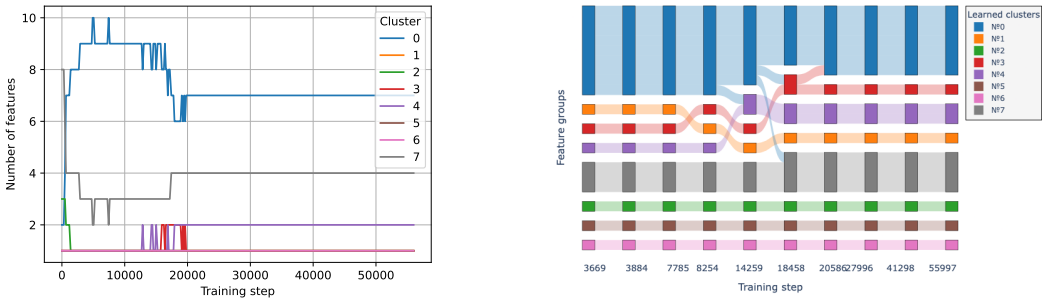
(*a*) Cluster sizes (*b*) Sankey diagram

Figure 12: Dynamics of feature assignments to clusters for Fuzzy K-means model trained on MIMIC for decompensation prediction. The features eventually end up assigned to a single cluster, replicating the best previously known grouping.



(*a*) Cluster sizes (*b*) Sankey diagram

Figure 13: Dynamics of feature assignments to clusters for K-means model trained on MIMIC for mortality prediction. Grouping does not change from the initialization.



(*a*) Cluster sizes (*b*) Sankey diagram

Figure 14: Dynamics of feature assignments to clusters for Fuzzy K-means model trained on MIMIC for mortality prediction. Grouping gradually evolves, stabilizing at around half point of the training.

Table 13: Prior grouping of HiRID features by organ system.

Group name	Features
Central nervous (Cns)	GCS Antwort, GCS Motorik, GCS Augenöffnen, RASS, ICP, TOF, Benzodiacepine, Alpha 2 Agonisten, Barbiturate, Propofol, Liquor/h, Nimodipin, Opiate, Non-opioide, NSAR, Ketalar, Peripherial Anesthesia, Antiepileptica, Anti delirant medi, Psychopharma, Muskelrelaxans, Anexate, Naloxon, Parkinson Medikaiton, pH Liquor, Laktat Liquor, Glucose Liquor
Circulatory	HR, T Central, ABPs, ABPd, ABPm, NIBPs, NIBPd, NIBPm, PAPm, PAPs, PAPd, PCWP, CO, SvO2(m), ZVD, ST1, ST2, ST3, Rhythmus, IN, OUT, Incrys, Incolloid, packed red blood cells, FFP, platelets, coagulation factors, norepinephrine, epinephrine, dobutamine, milrinone, levosimendan, theophyllin, vasopressin, desmopressin, vasodilatators, ACE Inhibitors, Sartane, Ca Antagonists, B-Blocker, Andere, Adenosin, Digoxin, Amiodaron, Atropin, Thyrotardin, Thyroxin, Thyreostatikum, Mineralokortikoid, Antihistaminka, Terlipressin, Troponin-T, creatine kinase, creatine kinase-MB, BNP, TSH, AMYL-S
Hematologic	Glucose Administration, Insuling Langwirksam, Insulin Kurzwirksam, Thrombozytenhemmer, Heparin, NMH, Others in Case of HIT, Marcoumar, Protamin, Anti Fibrinolyticum, Lysetherapie, Pankreas Enzyme, VitB Substitution, Weight, a-BE, a COHb, a Hb, a Lac, a MetHb, v-Lac, aPTT, Fibrinogen, FII, Factor V, Factor VII, factor X, INR, albumin, glucose, Ammoniak, Hb, total white blood cell count, platelet count, MCH, MCHC, MCV, Ferritin, Lipase
Immune	Administriation of antibiotics, Administation of antimycotic, administration of antiviral, Antihelmenticum, Steroids, Enteral Feeding, steroids, non-steroids, Chemotherapie, Immunoglobulin, Immunsuppression, GCSF, C-reactive protein, procalcitonin, lymphocyte, Neutr, Segm. Neut., Stabk. Neut., BSR, Cortisol
Hepatic	ASAT, ALAT, bilirubine, total, Bilirubin, direct, alkaline phosphatase, gamma-GT
Pulmonary	SpO2, ETCO2, RR, supplemental oxygen, FIO2, Peep, Ventilator mode, TV, Spitzendruck, Plateaudruck, AWPmean, RR set, AiwayCode, Beh. Pulm. Hypertonie, a pCO2, a PO2, a SO2, Zentral venöse sättigung, pH Drain, AMYL-Drainag
Renal	OUTurine/h, K-sparend, Aldosteron Antagonist, Loop diuretics, Thiazide, Acetazolamide, Haemofiltration, Parenteral Feeding, Kalium, Phosphat, Na, Mg, Ca, Trace elements, Bicarbonate, a HCO3-, a pH, K+, Na+, Cl-, Ca2+ ionizied, Ca2+ total, phosphate, Mg lab, Urea, creatinine, urinary creatinin, urinary Na+, urinary urea

Table 14: Prior grouping of MIMIC-III features by organ system.

Group name	Features
Central nervous (Cns)	Glasgow coma scale eye opening, Glasgow coma scale motor response, Glasgow coma scale total, Glasgow coma scale verbal response
Circulatory	Diastolic blood pressure, Heart Rate, Mean blood pressure, Systolic blood pressure, Temperature, Capillary refill rate
Hematologic	Glucose
Pulmonary	Fraction inspired oxygen, Oxygen saturation, Respiratory rate
Renal	pH
Other	Time, Height, Weight

Table 15: Prior grouping of HiRID features by measurement type.

Group name	Features
Derived from raw data	ETCO2, OUTurine/h, IN, OUT, Incrys, Incolloid
Laboratory	a-BE, a COHb, a Hb, a HCO3-, a Lac, a MetHb, a pH, a pCO2, a PO2, a SO2, Zentral venöse sättigung, Troponin-T, creatine kinase, creatine kinaseMB, v-Lac, BNP, K+, Na+, Cl-, Ca2+ ionizied, Ca2+ total, phosphate, Mg lab, Urea, creatinine, urinary creatinin, urinary Na+, urinary urea, ASAT, ALAT, bilirubine, total, Bilirubin, direct, alkaline phosphatase, gamma-GT, aPTT, Fibrinogen, FII, Factor V, Factor VII, factor X, INR, albumin, glucose, Ammoniak, C-reactive protein, procalcitonin, lymphocyte, Neutr, Segm. Neut., Stabk. Neut., BSR, Hb, total white blood cell count, platelet count, MCH, MCHC, MCV, Ferritin, TSH, AMYL-S, Lipase, Cortisol, pH Liquor, Laktat Liquor, Glucose Liquor, pH Drain, AMYL-Drainag
Monitored	HR, T Central, ABPs, ABPd, ABPm, NIBPs, NIBPd, NIBPm, PAPm, PAPs, PAPd, PCWP, CO, SvO2(m), ZVD, ST1, ST2, ST3, SpO2, ETCO2, RR, ICP, TOF, FIO2, Peep, Ventilator mode, TV, Spitzendruck, Plateaudruck, AWPmean, RR set
Observed	ZVD, Rhythmus, supplemental oxygen, GCS Antwort, GCS Motorik, GCS Augen öffnen, RASS, ICP, AiwayCode, Haemofiltration, Liquor/h, Weight
Treatment	packed red blood cells, FFP, platelets, coagulation factors, norepinephrine, epinephrine, dobutamine, milrinone, levosimendan, theophyllin, vasopressin, desmopressin, vasodilatators, ACE Inhibitors, Sartane, Ca Antagonists, BBlocker, Andere, Adenosin, Digoxin, Amiodaron, Atropin, K-sparend, Aldosteron Antagonist, Loop diuretics, Thiazide, Acetazolamide, Administriation of antibiotics, Administation of antimycotic, administration of antiviral, Antihelmenticum, Benzodiacepine, Alpha 2 Agonisten, Barbiturate, Propofol, Glucose Administration, Insuling Langwirksam, Insulin Kurzwirksam, Nimodipin, Opiate, Non-opioide, NSAR, Ketalar, Peripherial Anesthesia, Steroids, Thrombozytenhemmer, Enteral Feeding, Parenteral Feeding, Heparin, NMH, Others in Case of HIT, Marcoumar, Protamin, Anti Fibrinolyticum, Kalium, Phosphat, Na, Mg, Ca, Trace elements, Bicarbonate, Antiepileptica, Anti delirant medi, Psychopharma, steroids, nonsteroids, Thyrotardin, Thyroxin, Thyreostatikum, Mineralokortikoid, Antihistaminka, Chemotherapie, Lysetherapie, Muskelrelaxans, Anexate, Naloxon, Beh. Pulm. Hypertonie, Pankreas Enzyme, Terlipressin, Immunoglobulin, Immunsuppression, VitB Substitution, Parkinson Medikaiton, GCSF

Table 16: Prior grouping of MIMIC-III features by measurement type.

Group name	Features
Laboratory	Glucose, pH
Monitored	Diastolic blood pressure, Heart Rate, Mean blood pressure, Systolic blood pressure, Temperature, Fraction inspired oxygen, Oxygen saturation, Respiratory rate
Observed	Glasgow coma scale eye opening, Glasgow coma scale motor response, Glasgow coma scale total, Glasgow coma scale verbal response, Capillary refill rate
Other	Time, Height, Weight

Table 17: Ablation results over various parameters for GMM (soft) trained on HiRID for circulatory failure prediction.

(*a*) Initialization

Parameter	AUPRC
k-means++	0.403 ± 0.173
prior	0.366 ± 0.017

(*b*) Unification

Parameter	AUPRC
bias	0.395 ± 0.173
bias_avg_linear	0.403 ± 0.005
bias_ext_catzero	0.366 ± 0.017
bias_sum_linear	0.396 ± 0.008

(*c*) Merge

Parameter	AUPRC
attention	0.366 ± 0.017
mean	0.403 ± 0.173

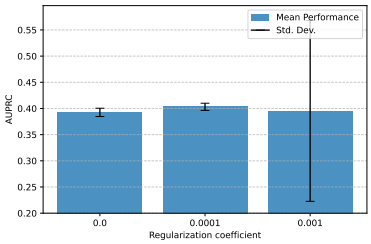
(*d*) Regularization type

Parameter	AUPRC
hard	0.393 ± 0.173
soft	0.403 ± 0.008

(*e*) Regularization coefficient

Parameter	AUPRC
0.0000	0.393 ± 0.008
0.0001	0.403 ± 0.007
0.0010	0.395 ± 0.173

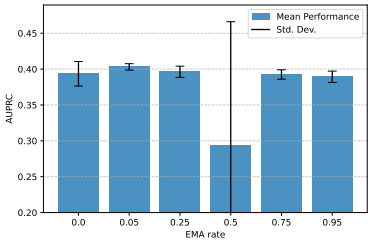
(*f*) Regularization coefficient



(*g*) EMA rate

Parameter	AUPRC
0.00	0.393 ± 0.017
0.05	0.403 ± 0.004
0.25	0.396 ± 0.008
0.50	0.293 ± 0.173
0.75	0.393 ± 0.007
0.95	0.389 ± 0.008

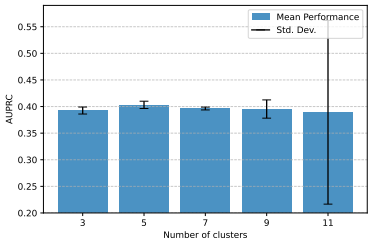
(*h*) EMA rate



(*i*) Number of clusters

Parameter	AUPRC
3	0.393 ± 0.007
5	0.403 ± 0.007
7	0.396 ± 0.003
9	0.395 ± 0.017
11	0.389 ± 0.173

(*j*) Number of clusters



(*k*) Membership threshold

Parameter	AUPRC
0.80	0.393 ± 0.008
0.85	0.395 ± 0.173
0.90	0.403 ± 0.017

(*l*) Membership threshold

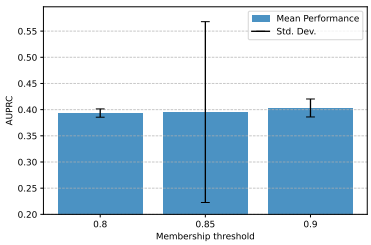


Table 18: Ablation results over various parameters for K-means trained on HiRID for mortality prediction.

(*a*) Initialization

Parameter	AUPRC
k-means++	0.631 ± 0.080
prior	0.615 ± 0.026

(*b*) Unification

Parameter	AUPRC
bias	0.631 ± 0.080
bias_avg_linear	0.615 ± 0.009
bias_ext_catzero	0.584 ± 0.078
bias_sum_linear	0.602 ± 0.014

(*c*) Merge

Parameter	AUPRC
attention	0.631 ± 0.080
mean	0.615 ± 0.078

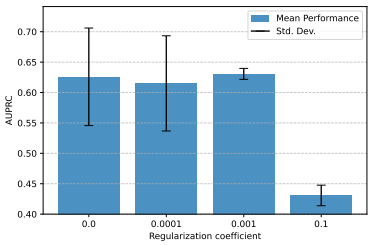
(*d*) Regularization type

Parameter	AUPRC
hard	0.631 ± 0.080
soft	0.602 ± 0.078

(*e*) Regularization coefficient

Parameter	AUPRC
0.0000	0.626 ± 0.080
0.0001	0.615 ± 0.078
0.0010	0.631 ± 0.009
0.1000	0.431 ± 0.017

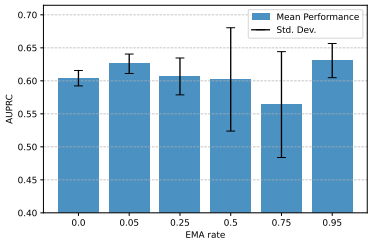
(*f*) Regularization coefficient



(*g*) EMA rate

Parameter	AUPRC
0.00	0.604 ± 0.012
0.05	0.626 ± 0.015
0.25	0.607 ± 0.028
0.50	0.602 ± 0.078
0.75	0.564 ± 0.080
0.95	0.631 ± 0.026

(*h*) EMA rate



(*i*) Number of clusters

Parameter	AUPRC
3	0.626 ± 0.080
5	0.631 ± 0.009
8	0.534 ± 0.078

(*j*) Number of clusters

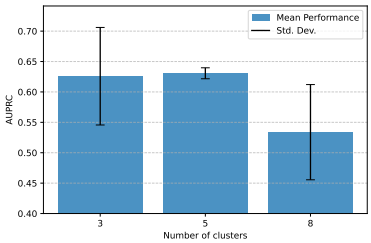


Table 19: Ablation results over various parameters for Fuzzy K-means trained on HiRID for mortality prediction.

(*a*) Initialization

Parameter	AUPRC
k-means++	0.597 ± 0.040
prior	0.629 ± 0.059

(*b*) Unification

Parameter	AUPRC
bias	0.608 ± 0.012
bias_avg_linear	0.610 ± 0.040
bias_ext_catzero	0.620 ± 0.059
bias_sum_linear	0.629 ± 0.016

(*c*) Merge

Parameter	AUPRC
attention	0.620 ± 0.059
mean	0.629 ± 0.040

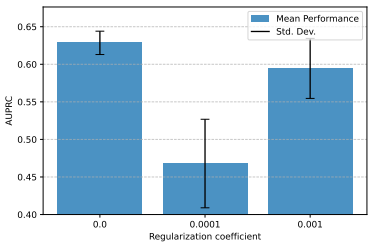
(*d*) Regularization type

Parameter	AUPRC
hard	0.629 ± 0.040
soft	0.610 ± 0.059

(*e*) Regularization coefficient

Parameter	AUPRC
0.0000	0.629 ± 0.016
0.0001	0.468 ± 0.059
0.0010	0.595 ± 0.040

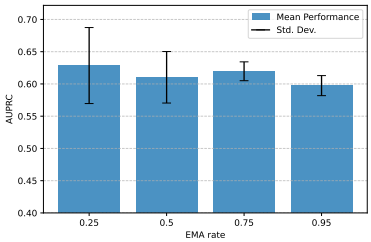
(*f*) Regularization coefficient



(*g*) EMA rate

Parameter	AUPRC
0.25	0.629 ± 0.059
0.50	0.610 ± 0.040
0.75	0.620 ± 0.015
0.95	0.597 ± 0.016

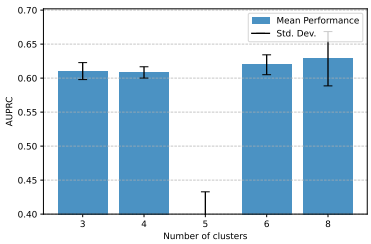
(*h*) EMA rate



(*i*) Number of clusters

Parameter	AUPRC
3	0.610 ± 0.012
4	0.608 ± 0.008
5	0.374 ± 0.059
6	0.620 ± 0.015
8	0.629 ± 0.040

(*j*) Number of clusters



(*k*) Fuzzy coefficient

Parameter	AUPRC
2	0.610 ± 0.012
5	0.629 ± 0.059
10	0.597 ± 0.016

(*l*) Fuzzy coefficient

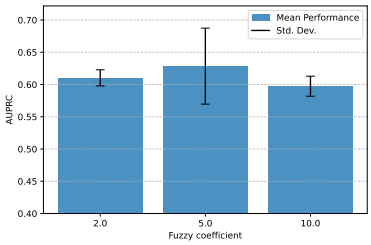


Table 20: Ablation results over various parameters for GMM trained on MIMIC for decompensation prediction.

(*a*) Initialization

Parameter	AUPRC
k-means++	0.378 ± 0.025
prior	0.391 ± 0.006

(*b*) Unification

Parameter	AUPRC
bias	0.385 ± 0.005
bias_avg_linear	0.391 ± 0.012
bias_ext_catzero	0.364 ± 0.025
bias_sum_linear	0.377 ± 0.011

(*c*) Merge

Parameter	AUPRC
attention	0.391 ± 0.025
mean	0.381 ± 0.012

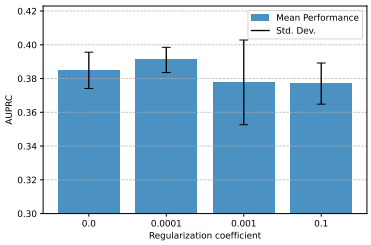
(*d*) Regularization type

Parameter	AUPRC
hard	0.380 ± 0.012
soft	0.391 ± 0.025

(*e*) Regularization coefficient

Parameter	AUPRC
0.0000	0.385 ± 0.011
0.0001	0.391 ± 0.007
0.0010	0.378 ± 0.025
0.1000	0.377 ± 0.012

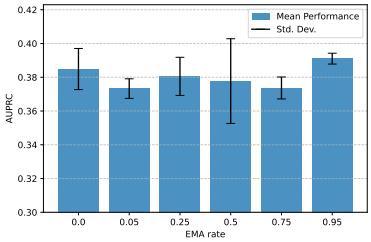
(*f*) Regularization coefficient



(*g*) EMA rate

Parameter	AUPRC
0.00	0.385 ± 0.012
0.05	0.373 ± 0.006
0.25	0.381 ± 0.011
0.50	0.378 ± 0.025
0.75	0.374 ± 0.007
0.95	0.391 ± 0.003

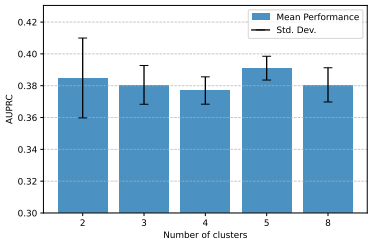
(*h*) EMA rate



(*i*) Number of clusters

Parameter	AUPRC
2	0.385 ± 0.025
3	0.381 ± 0.012
4	0.377 ± 0.009
5	0.391 ± 0.007
8	0.381 ± 0.011

(*j*) Number of clusters



(*k*) Covariance type

Parameter	AUPRC
diagonal	0.381 ± 0.011
full	0.391 ± 0.012
spherical	0.385 ± 0.025
tied	0.381 ± 0.004

Table 21: Ablation results over various parameters for K-means trained on MIMIC for mortality prediction.

(*a*) Initialization

Parameter	AUPRC
k-means++	0.525 ± 0.106
prior	0.516 ± 0.068

(*b*) Unification

Parameter	AUPRC
bias	0.520 ± 0.014
bias_avg_linear	0.520 ± 0.015
bias_ext_catzero	0.503 ± 0.068
bias_sum_linear	0.525 ± 0.106

(*c*) Merge

Parameter	AUPRC
attention	0.515 ± 0.012
mean	0.525 ± 0.106

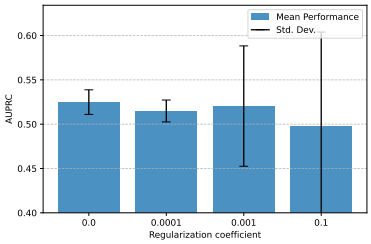
(*d*) Regularization type

Parameter	AUPRC
hard	0.520 ± 0.015
soft	0.525 ± 0.106

(*e*) Regularization coefficient

Parameter	AUPRC
0.0000	0.525 ± 0.014
0.0001	0.515 ± 0.012
0.0010	0.520 ± 0.068
0.1000	0.498 ± 0.106

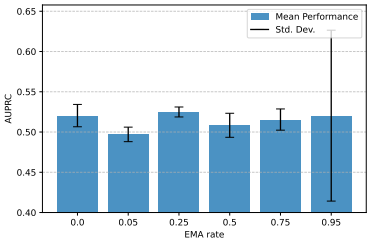
(*f*) Regularization coefficient



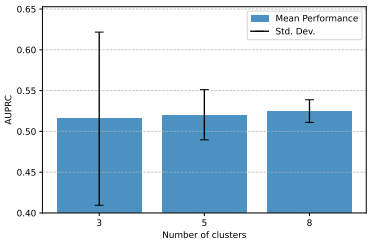
(*g*) EMA rate

Parameter	AUPRC
0.00	0.520 ± 0.014
0.05	0.497 ± 0.009
0.25	0.525 ± 0.006
0.50	0.508 ± 0.015
0.75	0.515 ± 0.013
0.95	0.520 ± 0.106

(*h*) EMA rate



(*i*) Number of clusters

Parameter	AUPRC
3	0.516 ± 0.106
5	0.520 ± 0.031
8	0.525 ± 0.014

(*j*) Number of clusters