Dynamic Outcomes-Based Clustering of Disease Trajectory in Mechanically Ventilated Patients

Emma Rocheteau[*1], **Ioana Bica**[2], **Pietro Liò**[1], **Ari Ercole**[3]

[1] Department of Computer Science and Technology, University of Cambridge, UK
[2] Department of Engineering Science, University of Oxford, UK
[3] Department of Anaesthesia, University of Cambridge, UK
*ecr38@cam.ac.uk

Abstract

The advancement of Electronic Health Records (EHRs) and machine learning have enabled a data-driven and personalised approach to healthcare. One step in this direction is to uncover patient sub-types with similar disease trajectories in a heterogeneous population. This is especially important in the context of mechanical ventilation in intensive care, where mortality is high and there is no consensus on treatment. In this work, we present a new approach to clustering mechanical ventilation episodes, using a multi-task combination of supervised, self-supervised and unsupervised learning techniques. Our dynamic clustering assignment is explicitly guided to reflect the *phenotype*, *trajectory* and *outcomes* of the patient. Experimentation on a real-world dataset is encouraging, and we hope that we could someday translate this into actionable insights in guiding future clinical research.

Introduction and Related Work

Patients on mechanical ventilation are a highly heterogeneous group, with widely differing outcomes. Some have relatively healthy lungs e.g. if they are recovering from surgery on another organ; whereas others have varying degrees of pulmonary failure. Pulmonary failure can be acute e.g. Acute Respiratory Distress Syndrome (ARDS) and deteriorate rapidly, or chronic, typically evolving slowly. Unfortunately, patients on ventilators have high mortality (Máca et al. 2017; Poole et al. 2017) and there is no established consensus on optimal treatment strategies from randomised controlled trials (Bein et al. 2016). Therefore, there is *great potential benefit to be gained from phenotype discovery* in order to guide future clinical studies.

To this end, we have developed a dynamic clustering approach for mechanically ventilated patients in the ICU. Previous work using simple clustering techniques has revealed *actionable* sub-phenotypes by secondary analysis of RCT data. For example, latent trajectory modelling of inflammatory biomarkers has revealed sub-types of ARDS (Famous et al. 2017). Clustering of transcriptomic data has revealed patient populations in which steroid therapy may be beneficial in sepsis (Antcliffe et al. 2019). Routinely collected data has also been used to find trajectory clusters in sepsis based on physiological parameters (Bhavani et al. 2022).

We know that temporal neural network architectures can handle the heterogeneous population in the ICU, both using supervised (Rocheteau, Liò, and Hyland 2021; Harutyunyan et al. 2019) and unsupervised (Miotto et al. 2016; Wang et al. 2020) approaches. Temporal clustering approaches have been applied successfully to other domains e.g. in Parkinson's (Zhang et al. 2019), diabetes (Rusanov, Prado, and Weng 2016) and cystic fibrosis (Lee and van der Schaar 2020) and increasingly in intensive care as discussed above.

We have designed our clusters to share similarities in *phenotype*, *trajectory* and *outcomes*. We generate a cluster for each hour of a patient's stay, meaning that if an event happens which alters the predicted trajectory and outcomes, there will be a shift in the cluster assignment. This is interesting, not only because it can reveal which events are associated with these shifts, but also what might have happened if the ventilation strategy had been different. We hope that our work could someday translate into actionable insights in guiding future clinical research.

Methods

Broadly, our strategy was to train a temporal encoder to embed the patient data at every timestep (this is analogous to returning all the hidden states for an LSTM model). We used a mixture of supervised, unsupervised and self-supervised learning to do this (see 'Prediction Tasks' below). Once the encoder training was complete, we used an unsupervised method to cluster the embeddings, so that we get a cluster for every timestep in the patient's ventilation episode. The code can be found at: https://github.com/EmmaRocheteau/Mechanical-Ventilation-Clustering.

The data consisted of both timeseries and static features. The supervised tasks included two binary tasks: predicting hospital mortality and the risk of receiving a tracheostomy[1], and two duration tasks: the remaining length of stay (LoS) from timestep t, and their remaining ventilation duration (VD). This ensured that the patient *outcomes* are stored within the embedding. In addition, we trained a decoder to reconstruct timestep t and the static data. This unsupervised approach encourages the embedding to retain the patient *phenotype*. Finally, we predicted timestep $t+1$, a self-supervised

[1] A tracheostomy is a procedure designed for long term mechanical ventilation of a patient.

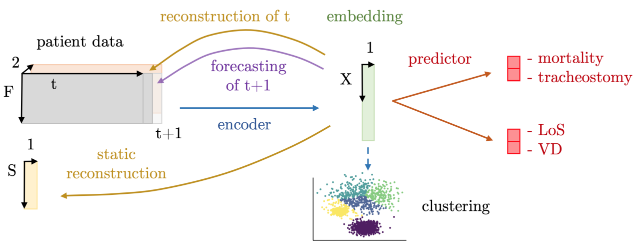

Figure 1: Overview of our model. Only one timestep, t, is shown for simplicity. F and S are the number of time series and static variables respectively. At timestep t, the static variables (yellow) and preceding time series variables (grey) and their corresponding decay indicator variables (orange, explained under ‘Time Series’ in Data Preprocessing in the Appendix) are given to the encoder, which produces an embedding (green) for timestep t. This is then given to the decoder networks (yellow), forecasting network (purple) and the predictor network to obtain the four patient outcomes (red). After training is complete, the test embeddings are used for clustering.

approach designed to embed the patient *trajectory*. See Figure 1 for a schematic.

Encoder In recent years, LSTMs have been by far the most popular model for predicting clinical outcomes and have achieved state-of-the-art results (Harutyunyan et al. 2019; Sheikhalishahi, Balaraman, and Osmani 2019; Rajkomar et al. 2018; Tong et al. 2022). They have also been applied to other patient prediction tasks e.g. forecasting diagnoses and medications (Choi et al. 2015; Lipton et al. 2015), and mortality prediction (Che et al. 2018; Harutyunyan et al. 2019; Shickel et al. 2019). More recently, the Transformer model (Vaswani et al. 2017) has marginally outperformed the LSTM when predicting LoS (Song et al. 2018). Rocheteau, Liò, and Hyland (2021) showed that Temporal Pointwise Convolution (TPC) outperformed both the LSTM and Transformer models on mortality and LoS. Therefore, we chose to investigate these three encoders. Details of their implementation are given in the Appendix.

K-Medoids Clustering We used k-medoids clustering to cluster the learned embeddings. K-medoids is similar to k-means, except that it operates with medoids rather than centroids. This means that the medoids will always be a true observation in the data, while that is not usually the case for centroids. The main advantage is that k-medoids are less sensitive to outliers than k-means, which is more suitable in this context where the data is noisy and heavily skewed[2].

Both k-means and k-medoids operate on pairwise similarities. We decided to use Euclidean distance rather than cosine similarity. This is because intuitively, it is not only the direction that the patient is moving in that matters, but also the distance along that axis. For example, if a particular ‘direction’ represents acute decompensated heart failure, we also care how severe the decompensation is.

We applied batch normalisation (Ioffe and Szegedy 2015) to the embeddings, to ensure that the embedding distribution remained within a reasonable range. The value of k (5 for all models) was chosen using the elbow method (see ‘Number of Clusters’ in the Appendix).

Data

We used the Amsterdam UMC database version 1.0.2 (Thoral et al. 2021), which contains 23,106 ICU admissions from 20,109 patients admitted between 2003 and 2016. We selected all of the mechanical ventilation episodes with a minimum duration of 4 hours, capping the maximum duration after 21 days to reduce computational costs. This corresponded to 14,836 episodes which occurred during 13,502 ICU admissions from a cohort of 12,597 unique patients. We selected 31 time series features and 14 static features. The data were split such that 70%, 15% and 15% were used for training, validation and testing respectively. These were split by *patient*, not ventilation episode, to avoid data leakage from the train set. Further details of the data are provided in the Appendix.

Prediction Tasks

Remaining Length of Stay and Ventilation Duration We assigned a remaining length of stay (LoS) and remaining ventilation duration (VD) target to each hour of the ventilation episode, ending when the patient dies or is extubated. We only trained on data from the first 21 days of the ventilation episode to protect against batches becoming overly long and slowing down training.

The remaining LoS and VD each have a significant positive skew which makes the duration tasks more challenging. We partly circumvent this by replacing the commonly used mean squared error (MSE) loss with mean squared *log* error (MSLE), as in Rocheteau, Liò, and Hyland (2021). We reported on 2 LoS and VD metrics: mean absolute deviation (MAD) and mean squared log error (MSLE). The MAD was used as the primary metric in Harutyunyan et al. (2019) but MSLE is arguably the more holistic metric (Rocheteau, Liò, and Hyland 2021).

Mortality and Tracheostomy Unlike the duration tasks, these tasks are static, i.e. the labels do not change during the ventilation episode. Both tasks have significant class imbalance (only 14.6% and 7.4% of patients died or received a tracheostomy respectively). In order to encourage the model to prioritise learning these important outcomes, we applied class weighting to the task. We used binary crossentropy as the loss function. We report the area under the receiver operating characteristic curve (AUROC) and the area under the precision recall curve (AUPRC) as metrics.

Reconstruction and Forecasting As shown in Figure 1, we use the embedding to reconstruct the timestep t, and forecast one timestep $(t + 1)$ ahead. For the reconstruction of t and forecast of $t + 1$, we apply the mean squared error. We also reconstruct the following static features: sex, urgency of admission, agegroup, weightgroup, and heightgroup. The

[2] Preliminary experiments revealed that k-means were more likely to produce small clusters which lay far away from the rest of the data, because it is more affected by outliers. This made the clustering process less reliable and reproducible.

first two are binary, and so we apply the binary crossentropy loss function. The other three are ordered categorical (as explained in 'Static Features' in the Appendix), therefore we use the mean squared error loss function. Since these tasks are *auxiliary* (we are not interested in the performance as an outcome of the model), we reported their loss function values as 'metrics' since they do not need to be interpretable.

The relative weightings of all of these tasks are given under 'Hyperparameter Search Methodology' in the Appendix.

Results

In this section, we highlight important performance differences between the three encoders, analyse an ablation study on the tasks, and provide a detailed analysis of the clusters produced by the TPC model. A deeper evaluation of the results can be found in the discussion.

Task Performance

(a) – Full Task Setting The TPC model performs significantly better than the LSTM and Transformer on the outcome tasks (Table 1a), which is in line with previous findings in MIMIC-IV and eICU (Rocheteau, Liò, and Hyland 2021). The superiority of the TPC model is also evident in the variational and ablation experiments. Interestingly, the Transformer performs poorly on the binary tasks but better on the duration tasks with respect to the LSTM. Additionally, the LSTM performs the best on the reconstruction and forecasting tasks (Table 14a). Possible reasons for these findings are explored in the discussion.

(b) – Variational Embedding Spaces We experimented with making the embeddings 'variational', by representing the embedding as a set of means and standard deviations to allow sampling of embedding coordinates. The rationale was that by forcing the embedding space to be smoother, we might improve the quality of the clustering as the distances between patients in the embedding space become more reliable. However, this was found to universally hurt performance (Table 1b and Table 14b) and it produced clusters which were more homogeneous in terms of outcomes and features, which was counter to the aim of producing clinically distinct clusters.

Ablation Study

We performed an ablation study on the tasks used to train the representation space. The full set of results and analysis are included in the Appendix. However, the trend is such that the best results for all tasks (except for the duration tasks) are achieved when all tasks are included (Table 5). The reason for the exception is in the duration only task setting (g), is explored further in the discussion. Overall, our ablation study indicates that having multiple competing learning objectives has a stabilising effect on learning the representation.

Cluster Analysis

As the best performing encoder, we have focused on analysing the clusters produced by the TPC model. In order to analyse the average differences between the patients in each cluster, it was necessary to flatten the clustering into one 'primary' cluster per patient. This was to prevent confusion, since patients can enter multiple clusters during their ICU stay (sometimes only for one or two timepoints), and this is disproportionately true of the long stay patients. The cluster in which each patient spent the majority of their time in was assigned its primary cluster. If there were multiple modes, then the mode experienced later in the sequence was chosen. The next two sections characterise the behaviour of the primary clusters. Subsequently, we analyse the dynamic aspects of the clustering from multiple different perspectives.

Differences in Phenotype and Outcomes Table 2 shows the mean outcomes for each cluster. We also analysed some key features in the original data, to visualise differences in patient *phenotype* that the model identified. The average values of key features in patients divided by primary cluster are shown in Table 3. Broadly we can say that:

- Cluster 1 contains the sickest patients, with an average mortality of 72.0%. They are short stay patients with low rates of tracheostomy as most do not survive or stay long enough to require complex respiratory weaning. Table 3 shows they are primarily ventilated with 'mandatory' ventilation settings, meaning the machine is breathing for the patient. Furthermore, they have evidence of mechanical and functional damage to the lung parenchyma. This is in keeping with severe respiratory distress. We could describe this phenotype as a *'early, life-threatening pulmonary injury'* patient group.
- Cluster 2 display substantial mortality and severe pulmonary dysfunction like cluster 1. However this phenotype is characterised by very long LoS and VD, with consequent high rates of tracheostomy: this represents patients who are difficult to wean from mechanical ventilation. This might be described as a *'pulmonary critical illness'* phenotype.
- Cluster 3 have the best outcomes, with short LoS and low mortality. They are extubated without tracheostomy. This appears to be a *'short stay'* phenotype who require a brief period of organ support, perhaps after significant surgery.
- Cluster 4 have relatively low mortality but high rates of tracheostomy. Table 3 shows modest levels of respiratory failure and good lung compliance. Thus, whilst these patients are difficult to wean from mechanical ventilation (like cluster 2), this is due to factors that are not primarily related to pulmonary pathology. We could therefore describe them as a *'general critical illness'* phenotype.
- Cluster 5 shows a moderate to severe group, who are not as acutely unwell as cluster 1, but are still high-risk. From Table 3 we see that pulmonary injury is not a prominent feature so we could characterise these patients as *'early, life-threatening non-pulmonary injury'* patients.

Overall, the findings from Tables 2 and 3 show that there are clinically meaningful differences between the clusters. These can be visualised in Figure 2.

Medoid Analysis The medoids produced by the clustering algorithm are shown in Figure 3 and give a description of a representative patient in each cluster. Note that each

Table 1: Encoder performance on the prediction tasks averaged over 5 independent training runs. The error margins are 95% confidence intervals. For mortality and tracheostomy, higher AUROC and AUPRC is better; for LoS and VD, lower MAD and MSLE is better. (a) shows the full multi-task setting as shown in Figure 1, (b) is a variational alternative to the full task setting. Statistically significant differences are indicated by daggers († = $p < 0.05$, ‡ = $p < 0.001$). If the result is significantly better than the comparison models*, it is highlighted in **blue**, if it is significantly worse it is highlighted in **pink**. *In (a) the statistical testing compares the three model types, in (b) each model type is compared to its corresponding 'non-variational' model in table (a).

		In-Hospital Mortality		**Tracheostomy**		**Length of Stay**		**Vent. Duration**	
	Model	**AUROC**	**AUPRC**	**AUROC**	**AUPRC**	**MAD**	**MSLE**	**MAD**	**MSLE**
(a)	TPC	**0.833±0.010**†	**0.644±0.013**‡	**0.804±0.007**‡	**0.507±0.020**†	**7.20±0.13**‡	**0.359±0.010**‡	**3.24±0.07**‡	**0.210±0.008**‡
	Transformer	0.697±0.012	0.434±0.019	0.760±0.012	0.419±0.033	8.46±0.07	0.495±0.007	3.95±0.20	0.256±0.016
	LSTM	0.823±0.002	0.608±0.008	0.774±0.002	0.473±0.015	9.16±0.06	0.663±0.008	5.57±0.04	0.681±0.011
(b)	TPC	**0.807±0.006**‡	**0.584±0.014**‡	**0.775±0.008**‡	**0.437±0.012**‡	**9.06±0.10**‡	**0.555±0.018**‡	**4.42±0.03**‡	**0.347±0.006**‡
	Transformer	**0.660±0.023**†	**0.373±0.039**†	**0.714±0.020**‡	**0.353±0.018**†	**9.42±0.27**‡	**0.623±0.020**‡	**4.63±0.27**‡	**0.359±0.030**‡
	LSTM	**0.803±0.004**‡	**0.555±0.006**‡	**0.748±0.005**‡	**0.411±0.010**‡	**10.2±0.1**‡	**0.813±0.016**‡	**5.95±0.04**‡	**0.775±0.007**‡

Table 2: Average outcomes by cluster ± 95% confidence intervals for the TPC model. Each patient has been classified into a primary cluster, which is the cluster that they spent the majority of their time in. LoS and VD are shown in days.

Cluster	**Patients**	**Mortality (%)**	**Tracheostomy (%)**	**Length of Stay**	**Vent. Duration**
1	232	72.0±5.8	1.3±1.5	3.8±0.8	2.4±0.3
2	133	34.6±8.2	38.3±8.4	30.0±3.6	21.4±2.2
3	1,292	1.9±0.7	1.5±0.7	2.8±0.3	0.7±0.0
4	347	4.0±2.1	31.1±4.9	22.0±1.8	7.4±0.9
5	227	26.0±5.7	8.4±3.6	13.0±1.6	7.2±0.9

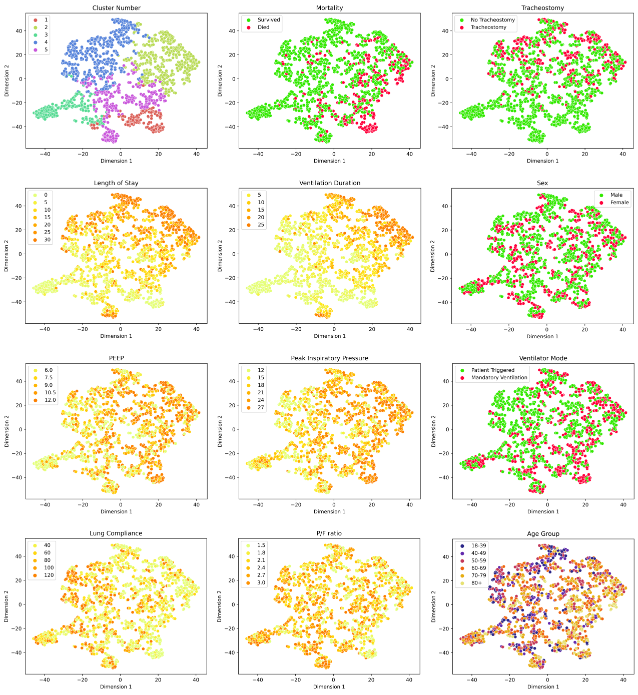

Figure 2: t-SNE plots for the embeddings produced by the TPC model. For these figures, 1500 random samples were selected from the test set and projected. In each plot, a different attribute has been highlighted.

medoid corresponds to a specific time-point in their ventilation episode.

- The medoid patient for cluster 1 (female, age 60-69) died 4 hours after the episode shown without a tracheostomy. Infection (high WBC) and pulmonary dysfunction are particularly noteworthy.
- The typical medoid patient representing cluster 2 (male, 80+ years old) received a tracheostomy 19 days after the episode shown, and was discharged at 23 days. This patient required late as well as early mandatory ventilation suggesting possible infectious complications (his CRP is also high).
- The medoid patient in cluster 3 (female, age 60-69) was discharged from hospital the day after her brief window of ventilation. She does not display substantial physiological derangement.
- The patient in cluster 4 (female, age 60-69) received a tracheostomy 3 days after the sequence shown. Her lung compliance and P/F ratio are both high, indicating good lung function. Therefore, we can conclude that she needed a tracheostomy for reasons other than lung injury.
- Lastly, the patient in cluster 5 (female, 80+ years old) stayed for 9 further days in hospital before being discharged. The short duration of ventilation and relatively normal pulmonary physiology is again consistent with a non-pulmonary phenotype.

Temporal Analysis Broadly, there are two perspectives when evaluating the dynamic aspects of this clustering.

Table 3: Key features averaged by cluster ± 95% confidence intervals. 'Urgency' is a flag given to the patient at admission. Mandatory Ventilation (MV) settings are provided in Table 13. The peak inspiratory pressure, P/F Ratio and PEEP are expressed in mmHg. A normal P/F ratio at sea level is ≈400-500mmHg; whereas 200-300mmHg is consistent with mild ARDS (Force et al. 2012). Lung compliance is expressed in ml/cmH$_2$O (normal for a mechanically ventilated patient is 50-100ml/cmH$_2$O).

Cluster	Age 70+ (%)	Sex (% male)	Urgency (%)	MV (%)	Peak Insp. Pressure	Lung Comp.	P/F Ratio	PEEP
1	52.2±6.5	59.7±6.3	63.4±6.3	68.3±0.8	25.3±0.2	32.7±0.5	217±2	10.09±0.07
2	54.1±8.6	65.8±8.1	39.1±8.4	43.2±0.4	23.2±0.1	36.8±0.3	220±1	9.97±0.03
3	39.8±2.7	69.7±2.5	14.9±1.9	38.6±0.6	16.1±0.1	58.8±0.7	260±1	6.78±0.03
4	25.9±4.6	68.4±4.9	41.5±5.3	22.1±0.4	17.8±0.1	57.5±0.4	237±1	8.19±0.28
5	40.1±6.4	69.6±5.9	43.2±6.5	41.8±0.5	20.3±0.1	47.1±0.4	243±1	8.83±0.38

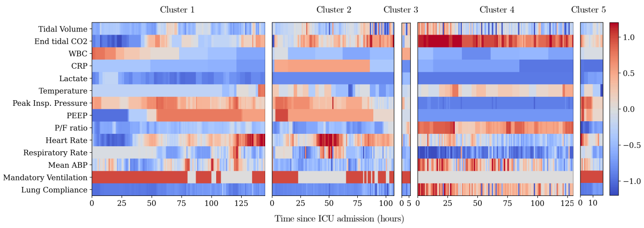

Figure 3: Raw data from each of the medoids. The data have been standardised around the mean value for each feature. Red means the value is high and blue means low. We can see that each medoid largely follows the average pattern for the cluster shown in Table 3. WBC is white blood count, CRP is C Reactive Protein, ABP is arterial blood pressure.

One is the 'Markovian' perspective, where we can examine the transition function between clusters. This is shown in Figure 4. Unsurprisingly, this reveals that the patient is always most likely to remain in the same cluster. However the most common inter-cluster transitions are from cluster 5 to cluster 4, and cluster 1 to cluster 5. Note that these clusters are next to one another and share lengthy borders in Figure 2. Most of the patients who transition to 'Died' come from cluster 1, and most of the 'Discharged' patients come from cluster 3.

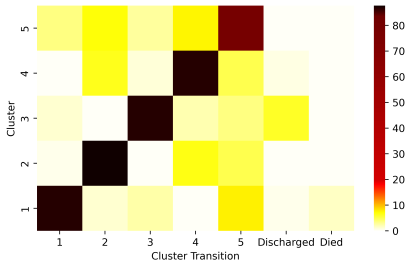

Figure 4: A transition matrix for the TPC model, showing the probability of entering each cluster at time $t + 1$, plus the categories 'discharged' or 'died', given their cluster at time t.

The other perspective is to look at the number of patients in each cluster at different time points after admission, and observe the transitions between them (Figure 5). Transitions from cluster 3 to 'extubated' are very common within the first day, but then they almost disappear by 3 days. This cannot be seen with the Markovian perspective in Figure 4. Cluster 2 contains patients with the longest ventilation episodes, which can be seen by its low rate of attrition over time.

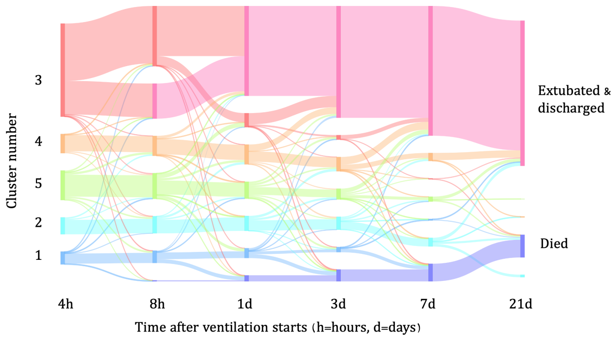

Figure 5: A sankey plot showing the evolution of the clustering across time. We begin at 4 hours to allow the clustering to stabilise at the start of the time series. At 21 days there are still some patients without a final outcome (mostly from cluster 2) but this is because they are ventilated for longer than 21 days and have been right censored.

Number of Clusters per Patient Figure 6 shows that most patients remain in only one cluster during their ventilation episode. However, when the distribution is broken down by primary cluster, we can see that this is heavily driven by the behaviour of cluster 3 patients, which tend to remain in cluster 3 for their entire ventilation duration (note that they tend to have short VDs so this is not so surprising). In contrast, clusters 2 and 5 most commonly appear alongside other clusters during a single ventilation episode. This means that for most episodes attributed to cluster 2 or 5, there are transitions either into or out of these clusters. These are explored next.

Cluster Transitions The clusters produced by the TPC model are remarkably stable over time, given that there is no explicit loss incentive to constrain the representation to behave in this way. Figure 7 shows the distribution of timepoints that the patients first enter their primary cluster. Clusters 2 and 3 are particularly likely to accurately assigned during the first hour of ventilation (87% and 89% respectively), while cluster 4 is the least likely to be identified early (64%).

Next, we investigated what we will refer to as 'stable' transitions between clusters. In order to be characterised as stable,

Table 4: Stable cluster transitions (origin cluster → destination cluster) with a count of $\geq$15, sorted by destination cluster. The median rather than the mean time is displayed to show a more representative time of transition (as there is positive skew).

Transition	Count	Median Time	Mortality (%)	Tracheostomy (%)	Urgency (%)	VD	LoS
3→1	17	3	76.5	0.0	47.1	0.5	0.7
5→1	29	16	51.7	10.3	55.2	4.3	5.3
1→3	28	11	10.7	0.0	67.9	1.0	2.6
5→3	46	9	15.2	4.3	41.3	1.2	6.5
2→4	28	17	10.7	21.4	42.9	6.2	12.8
5→4	27	10	11.1	7.4	48.1	3.4	9.1
1→5	25	3	44.0	4.0	68.0	3.9	6.5
3→5	15	4	13.3	13.3	53.3	1.9	4.6
4→5	15	56	26.7	26.7	46.7	6.6	11.5

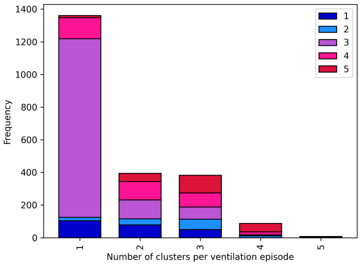

Figure 6: Distribution of the number of clusters that the patient enters during their ventilation episode, separated by *primary* cluster (shown by the colour key). For example, cluster 3 (purple) mainly appears on its own i.e. the patient starts the episode in cluster 3 and remains in cluster 3 for the whole duration, whereas cluster 5 (red) rarely appears on its own.

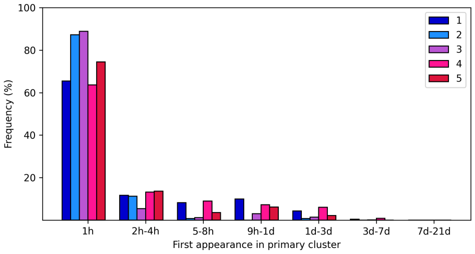

Figure 7: Percentage of patients who enter their primary cluster, by time since the start of the ventilation episode.

the origin cluster needed to remain stable in the 5 hours preceding the transition, and the patient was not permitted to re-enter the origin cluster for 5 hours following the transition. This was primarily to screen out patients who were at the boundary between two clusters, continually crossing back and forth but not representing a true transition from one to the other. Before screening, there were 22,036 cluster transitions, corresponding to 870 separate ventilation episodes (39% of the total in the test set). Of these transitions, only 291 represented stable movement between clusters. We further removed any transitions between two clusters that had fewer than 15 transition examples, as this would be insufficient to analyse. The remaining 230 transitions are shown in Table 4.

Firstly, it is noteworthy that the outcomes reflect the *destination* cluster, not the origin cluster. The exception to this is the 'urgency' column, which is not an outcome, but a label assigned at *admission* and hence is more likely to reflect the *origin* cluster (although it is worth mentioning that the origin cluster is not necessarily the cluster at admission).

Cluster 5 stands out as being disproportionately involved in inter-cluster transitions. Of these, the most common is 5→3, which occurs when the model overestimates the risk to the patient early on in the ventilation episode. Not shown in Figure 4, is that the average predicted risk of death drops from 56.4% 5 hours prior to the transition, to 41.7% at the point of transition. There is also a corresponding reduction in tracheostomy risk (-13%), LoS (-17.1% after adjustment[3]) and VD (-26.4% after adjustment) as predicted by the model, and dramatic improvements in physiological parameters such as lung compliance (+35%) and P/F ratio (+15%).

Another interesting transition is 3→1, which happens when the model initially believes the patient to be relatively healthy, but then quickly re-adjusts to predict poor outcomes. Looking in more detail at the raw data, we discovered that these patients are younger (only 23.5% are 70+), which could explain why the model was initially optimistic and why the deterioration is so rapid[4]. We also observed a deterioration in the lung compliance (-26.3%) and P/F ratios (-12.8%), and a change in the ventilator settings – namely higher PEEP and peak inspiratory pressure and lower tidal volumes – reflecting a drop in lung compliance of the patients. Most of these patients died within 12 hours of the transition to cluster 1.

Reliability We investigated the reproducibility of these phenotypes. We chose to analyse the clusters in the following settings: i) alternative encoder models, ii) retraining the TPC

[3] There is a 5 hour gap between these predictions, therefore this time difference needs to be removed from the first prediction.

[4] This is because younger patients can mask a problem by compensating deceptively well, until they reach a point where the homeostatic mechanisms can no longer cope.

model with different random seeds and iii) varying the value of k. The clusters were found to be surprisingly stable, with key features of the extracted phenotypes remaining similar between models. With increasing value of k, we noticed that rather than completely rearranging the position of the clusters, increasing k progressively subdivides existing clusters, hinting that the clusters are hierarchically organised (more in the discussion). The full analysis is included in the Appendix.

Discussion

We evaluated the use of TPC model, trained using supervised, unsupervised and self-supervised learning techniques, for the purposes of *phenotype* discovery in mechanically ventilated patients. We will discuss the most important findings in turn.

Firstly, we reaffirmed that the TPC model performs better than alternative encoders on EHR data for patient outcome prediction. This time on the Amsterdam UMC database (Thoral et al. 2021), and with added tasks.

Secondly, we found that the Transformer outperformed LSTM on LoS and VD, but performed much worse on the mortality task, and slightly worse on the tracheostomy task. This may be because the task weighting was more favourouble to the LSTM and TPC models, whereas the Transformer would have benefited from greater weighting towards the binary tasks. Another possibility is that the binary tasks benefit from biases in the LSTM and TPC encoders, because these models naturally emphasise recent timepoints (and these are especially important for solving the mortality task). As for the reason that the Transformer performs better on tracheostomy than mortality, it could be because there is positive correlation between the LoS, VD and tracheostomy tasks. Solving the duration tasks makes the tracheostomy task easier, whereas the relation to mortality is more complex (Figure 2).

To briefly comment on the reconstruction results in Table 14 in the Appendix; it may seem surprising that the LSTM model performs best on the reconstruction and forecasting tasks. However, this could be explained if the LSTM is creating simpler 'lower level' representations that are easier to translate back to the original data with the decoder networks.

Thirdly, Table 5 reveals a general trend that the more tasks that are added, the better results across all the tasks, with particular benefits to the tracheostomy task. The exception to this was the duration only setting. There are two possible explanations for the discrepancy:

1. The weighting of the duration task was not sufficient.
2. The tracheostomy task (but not mortality) reduces the performance on the duration tasks.

The former does not seem likely, because the Transformer is probably over-weighting the duration tasks, and yet, it follows the same trend as the LSTM and TPC. The latter may appear to be counter-intuitive, because the duration tasks are correlated with tracheostomy. Usually this is an advantage of multitask learning, because it enhances the signal:noise ratio when certain types of noise only apply to one task. However, looking closely at Figure 2, we can see that there is an area of patients near the bottom of the figure, in cluster 5. These patients have long VD and LoS but have been separated from the other long stay patients in clusters 2 and 4. The separation can be attributed to these patients never receiving tracheostomies, therefore the tracheostomy task forces the representation space to separate these groups when they would be otherwise be aligned. Given the simple nature of the predictor networks, this may harm the performance on the duration tasks because the predictor cannot effectively map these patients to appropriately long LoS. This theory could be formally tested by accompanying the duration tasks with the mortality task only.

Finally, regarding the repeatability of the clustering, we demonstrated that key aspects of the learned representations (both of different encoders and TPC instances) are consistently recognised. The separation on other traits, especially distinguishing the sickest patients from the moderately ill, was more malleable. This suggests that perhaps there is not a well defined distinction between these, but rather a scale of deterioration, through which an arbitrary line can be drawn.

Limitations and Future Work

Hierarchical Clustering It is evident that certain clusters are more related than others. A tree based hierarchy of clusters seems more natural than a flat structure. We are particularly interested in modifying an approach for genetics data (Patel et al. 2022; Chami et al. 2020; Corso et al. 2021), and are hoping to apply it to the ICU.

Contrastive Learning We are investigating the use of contrastive learning to regularise the embedding space (e.g. Yèche et al. (2021)). Currently, there is no explicit loss to enforce relative positioning of the embeddings. Despite this, we have empirically found the clusters to be very stable, both temporally and to encoder type. This is likely to be because our predictor and decoder networks are very simple, meaning that the embedding space does not have the freedom to model similar patient trajectories (which should be in the same cluster) in disparate parts of the embedding space. Nevertheless, contrastive learning could provide further regularisation.

Generalisability While we have shown that the clusters are surprisingly stable, repeating the work on another dataset (e.g. MIMIC-IV or eICU) would strengthen this assessment.

Summary

While we acknowledge important limitations in our work, we have shown that:

1. The TPC model outperforms alternative encoders on patient outcome prediction tasks.
2. We can generate clinically meaningful and interpretable clusters using this technique.
3. The phenotypes are similar across choices of encoder and number of clusters.
4. The cluster assignment is remarkably stable over time, and membership is determined early on. This is particularly encouraging as a substrate for future intervention studies, because they rely on phenotyping before any intervention.
5. Stable cluster transitions do occur but they are infrequent. Studying these transitions with a view towards understanding the causes is an important avenue for future work.

Acknowledgements

The authors would like to thank Petar Veličković, Sophie Xhonneux, Stephanie Hyland and Mihaela van der Schaar for helpful discussions and advice. We also thank the Armstrong Fund, the Frank Edward Elmore Fund, and the School of Clinical Medicine at the University of Cambridge for their generous funding.

References

Antcliffe, D. B.; Burnham, K. L.; Al-Beidh, F.; Santhakumaran, S.; Brett, S. J.; Hinds, C. J.; Ashby, D.; Knight, J. C.; and Gordon, A. C. 2019. Transcriptomic Signatures in Sepsis and a Differential Response to Steroids. From the VANISH Randomized Trial. *Am J Respir Crit Care Med*, 199(8): 980–986.

Bein, T.; Grasso, S.; Moerer, O.; Quintel, M.; Guerin, C.; Deja, M.; Brondani, A.; and Mehta, S. 2016. The standard of care of patients with ARDS: ventilatory settings and rescue therapies for refractory hypoxemia. *Intensive Care Medicine*, 42(5): 699–711.

Bhavani, S. V.; Semler, M.; Qian, E. T.; Verhoef, P. A.; Robichaux, C.; Churpek, M. M.; ; and Coopersmith, C. M. 2022. Development and validation of novel sepsis subphenotypes using trajectories of vital signs. *Intensive Care Med*, 48(11): 1582–1592.

Chami, I.; Gu, A.; Chatziafratis, V.; and Ré, C. 2020. From Trees to Continuous Embeddings and Back: Hyperbolic Hierarchical Clustering. *CoRR*, abs/2010.00402.

Che, Z.; Purushotham, S.; Cho, K.; Sontag, D.; and Liu, Y. 2018. Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Scientific Reports*, 8(1): 6085.

Choi, E.; Bahadori, M. T.; Schuetz, A.; Stewart, W. F.; and Sun, J. 2015. Doctor AI: Predicting Clinical Events via Recurrent Neural Networks. *JMLR workshop and conference proceedings*, 56: 301–318.

Corso, G.; Ying, R.; Pándy, M.; Veličković, P.; Leskovec, J.; and Liò, P. 2021. Neural Distance Embeddings for Biological Sequences.

Falcon, W. A. 2019. PyTorch Lightning. *GitHub*.

Famous, K. R.; Delucchi, K.; Ware, L. B.; Kangelaris, K. N.; Liu, K. D.; Thompson, B. T.; Calfee, C. S.; and Network, A. 2017. Acute Respiratory Distress Syndrome Subphenotypes Respond Differently to Randomized Fluid Management Strategy. *Am J Respir Crit Care Med*, 195(3): 331–338.

Force, A. D. T.; Ranieri, V. M.; Rubenfeld, G. D.; Thompson, B. T.; Ferguson, N. D.; Caldwell, E.; Fan, E.; Camporota, L.; and Slutsky, A. S. 2012. Acute Respiratory Distress Syndrome: The Berlin Definition. *JAMA*, 307(23): 2526–2533.

Harutyunyan, H.; Khachatrian, H.; Kale, D. C.; Ver Steeg, G.; and Galstyan, A. 2019. Multitask Learning and Benchmarking with Clinical Time Series Data. *Scientific Data*, 6(96).

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. *CoRR*, abs/1512.03385.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Ioffe, S.; and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, 448–456. JMLR.

Kalchbrenner, N.; Espeholt, L.; Simonyan, K.; van den Oord, A.; Graves, A.; and Kavukcuoglu, K. 2016. Neural Machine Translation in Linear Time. *CoRR*, abs/1610.10099.

Kingma, D. P.; and Ba, J. 2014. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980.

Lee, C.; and van der Schaar, M. 2020. Temporal Phenotyping using Deep Predictive Clustering of Disease Progression.

Lin, M.; Chen, Q.; and Yan, S. 2013. Network In Network. arXiv:1312.4400.

Lipton, Z. C.; Kale, D. C.; Elkan, C.; and Wetzel, R. C. 2015. Learning to Diagnose with LSTM Recurrent Neural Networks. *CoRR*, abs/1511.03677.

Miotto, R.; Li, L.; Kidd, B. A.; and Dudley, J. T. 2016. Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records. *Scientific Reports*, 6(1): 26094.

Máca, J.; Jor, O.; Holub, M.; Sklienka, P.; Burša, F.; Burda, M.; Janout, V.; and Ševčík, P. 2017. Past and Present ARDS Mortality Rates: A Systematic Review. *Respiratory Care*, 62(1): 113–122.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E. Z.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, 8024–8035. Curran Associates, Inc.

Patel, A.; Montserrat, D. M.; Bustamante, C.; and Ioannidis, A. 2022. Hyperbolic geometry-based deep learning methods to produce population trees from genotype data. *bioRxiv*.

Poole, J.; McDowell, C.; Lall, R.; Perkins, G.; McAuley, D. F.; Gao, F.; and Young, D. 2017. Individual patient data analysis of tidal volumes used in three large randomized control trials involving patients with acute respiratory distress syndrome. *BJA: British Journal of Anaesthesia*, 118(4): 570–575.

Rajkomar, A.; Oren, E.; Chen, K.; et al. 2018. Scalable and accurate deep learning with electronic health records. *Nature*, 1(1): 18.

Rocheteau, E.; Liò, P.; and Hyland, S. 2021. Temporal Pointwise Convolutional Networks for Length of Stay Prediction in the Intensive Care Unit. In *Proceedings of the Conference on Health, Inference, and Learning*, CHIL '21, 58–68. New York, NY, USA: Association for Computing Machinery.

Rocheteau, E.; Liò, P.; and Hyland, S. 2020. Predicting Length of Stay in the Intensive Care Unit with Temporal Pointwise Convolutional Networks.

Rusanov, A.; Prado, P. V.; and Weng, C. 2016. Unsupervised Time-Series Clustering Over Lab Data for Automatic Identification of Uncontrolled Diabetes. In *2016 IEEE International Conference on Healthcare Informatics (ICHI)*, 72–80.

Sheikhalishahi, S.; Balaraman, V.; and Osmani, V. 2019. Benchmarking Machine Learning Models on eICU Critical Care Dataset. arXiv:1910.00964.

Shickel, B.; Loftus, T. J.; Adhikari, L.; Ozrazgat-Baslanti, T.; Bihorac, A.; and Rashidi, P. 2019. DeepSOFA: A Continuous Acuity Score for Critically Ill Patients using Clinically Interpretable Deep Learning. In *Scientific Reports*.

Song, H.; Rajan, D.; Thiagarajan, J.; and Spanias, A. 2018. Attend and Diagnose: Clinical Time Series Analysis using Attention Models. In *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 4091–4098.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15: 1929–1958.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S. E.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2014. Going Deeper with Convolutions. *CoRR*, abs/1409.4842.

Thoral, P.; Driessen, R.; and Dam, T. 2020. AmsterdamUMCdb. *GitHub*.

Thoral, P. J.; Peppink, J. M.; Driessen, R. H.; Sijbrands, E. J. G.; Kompanje, E. J. O.; Kaplan, L.; Bailey, H.; Kesecioglu, J.; Cecconi, M.; Churpek, M.; Clermont, G.; van der Schaar, M.; Ercole, A.; Girbes, A. R. J.; and Elbers, P. W. G. 2021. Sharing ICU Patient Data Responsibly Under the Society of Critical Care Medicine/European Society of Intensive Care Medicine Joint Data Science Collaboration: The Amsterdam University Medical Centers Database (AmsterdamUMCdb) Example. *Critical Care Medicine*, 49(6).

Tong, C.; Rocheteau, E.; Veličković, P.; Lane, N.; and Liò, P. 2022. *Predicting Patient Outcomes with Graph Representation Learning*, 281–293. Cham: Springer International Publishing.

van den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A. W.; and Kavukcuoglu, K. 2016. WaveNet: A Generative Model for Raw Audio. *CoRR*, abs/1609.03499.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, u.; and Polosukhin, I. 2017. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, 6000–6010. Curran Associates Inc.

Wang, Y.; Zhao, Y.; Therneau, T. M.; Atkinson, E. J.; Tafti, A. P.; Zhang, N.; Amin, S.; Limper, A. H.; Khosla, S.; and Liu, H. 2020. Unsupervised machine learning for the discovery of latent disease clusters and patient subgroups using electronic health records. *Journal of Biomedical Informatics*, 102: 103364.

Yèche, H.; Dresdner, G.; Locatello, F.; Hüser, M.; and Rätsch, G. 2021. Neighborhood Contrastive Learning Applied to Online Patient Monitoring. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 11964–11974. PMLR.

Zhang, X.; Chou, J.; Liang, J.; Xiao, C.; Zhao, Y.; Sarva, H.; Henchcliffe, C.; and Wang, F. 2019. Data-Driven Subtyping of Parkinson's Disease Using Longitudinal Clinical Records: A Cohort Study. *Scientific Reports*, 9(1): 797.

Further Analyses

Ablation Study

We performed an ablation study on the tasks used to train the representation space. The results are shown in Table 5. Firstly, we see that the best results for all tasks (except for the duration tasks) are achieved in the full multi-task setting. Not a single metric improves in the other ablation settings, and yet at least one metric showed a deterioration in performance (the exception in task setting (g) is discussed below). Overall this indicates that having multiple competing learning objectives has a stabilising effect on learning the representation.

(c) – No Forecasting Experiment (c) included all the tasks except forecasting one timestep ahead. When we compare experiment (c) to (a), we see that the results are mostly similar, but there is a consistent decrease in performance, which is statistically significant at the $p<0.05$ level on the tracheostomy task (AUPRC in the TPC model and AUROC in the Transformer model). On the reconstruction task, again the performance is similar but statistically worse in the last timestep reconstruction in the LSTM model. This means that the forecasting task is contributing slightly to the performance in (a), but the benefit is small.

(d) – No Reconstruction Experiment (d) removes both the timestep t reconstruction and the static data reconstruction tasks, but keeps the forecasting task. The effect size is larger than in (c), but again is only statistically significant on the tracheostomy task. The forecasting task performs significantly worse in the Transformer and LSTM models without the reconstruction.

(e) – Prediction Tasks Only Experiment (e) includes the binary and duration prediction tasks, but no reconstruction or forecasting. The performance again deteriorates, particularly on the tracheostomy task, we also start to see a more noticeable deterioration in the duration tasks, although this is not yet statistically significant.

(f) – Binary Tasks Only Experiment (f) follows the trend of worsening performance as tasks are removed. This means that the mortality and tracheostomy tasks consistently benefit from supplementary tasks which help to distinguish signal from noise.

(g) – Duration Tasks Only Experiment (g) shows unexpected results; all of the models return better results when only predicting LoS and VD. This is not what has been observed previously in multitask settings ((Rocheteau, Liò, and Hyland 2021; Harutyunyan et al. 2019)). This is explored further in the discussion.

However, overall the trend is such that the more tasks that are included, the better the average results across tasks.

Cluster Reliability

Choice of Encoder Figure 8 compares the cluster assignments using different encoder models. It is encouraging that there is a strong cohesion between some of the clusters, meaning that the models are picking out genuine and consistent patterns in the data.

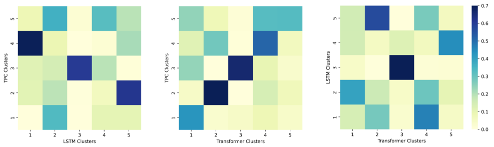

Figure 8: A comparison of the cluster assignments produced by different encoders. We can see that there is strong cohesion between some of the clusters. For example, cluster 3 appears to be the same in all three models.

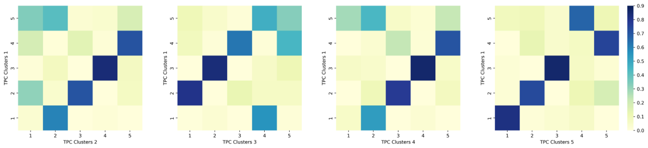

Figure 9: A comparison of the cluster assignments produced by TPC models which have been trained with different random seeds.

If we examine the TPC/LSTM comparison (far left in Figure 8) we can see that clusters 2, 3 and 4 in the TPC correspond to 5, 3 and 1 respectively in the LSTM. In addition, clusters 1 and 5 (TPC) imperfectly map to 2 and 4 (LSTM) – the main difference being that some additional patients in cluster 5 in the TPC map to cluster 2 in the LSTM. This means that the k-medoids algorithm has placed a different boundary between these groups in the clustering process. Looking back to Figure 2, clusters 1 and 5 are revealed to be neighbours – in fact, cluster 5 appears to envelope cluster 1, suggesting that 1 is a sub-cluster of 5. This is also consistent with the clinical picture shown in Table 2, where the main difference is that cluster 1 appear to have acute pulmonary dysfunction, most likely in addition to other organ failures. Therefore, the explanation for this discrepancy is that the LSTM has a more generous threshold than the TPC for inclusion in its highest risk *'early, life-threatening pulmonary injury'* category (cluster 2).

In the TPC/Transformer comparison, the clusters largely correlate, except that cluster 5 patients in the TPC have been placed into cluster 4 in the Transformer. Further investigation revealed these to be patients with increased risk of receiving a tracheostomy (i.e. the patients which lay closest to the decision boundary between the clusters).

The LSTM/Transformer comparison mirrors some of the correlations in TPC/LSTM but the mapping is less precise. This could be because the models do not perform as well, making the representations less reliable. It could also be because the models have different affinities for the various tasks, creating divergent biases in the representation space.

It is worth noting that in all three models, cluster 3 is the most distinct. This is unsurprising because it corresponds to patients whose physiology is closest to 'normal'. Therefore this group is the most homogeneous and can always be identified easily.

Retraining TPC We retrained the TPC model 5 times with different initialisation generated by different random seeds

Table 5: Prediction task results for the task ablation study. The full task setting from Table 1a has been repeated for ease of comparison. Various task ablations are compared to (a): (c) includes all tasks except for the forecasting task, (d) includes all tasks except for the reconstruction tasks, (e) includes only the prediction tasks, (f) is only the binary tasks, and (g) is only the duration tasks. The colour scheme, metrics and statistical test comparisons are explained in the legend to Table 1.

		In-Hospital Mortality		**Tracheostomy**		**Length of Stay**		**Vent. Duration**	
	Model	**AUROC**	**AUPRC**	**AUROC**	**AUPRC**	**MAD**	**MSLE**	**MAD**	**MSLE**
(a)	TPC	0.833±0.010	0.644±0.013	0.804±0.007	0.507±0.020	7.20±0.13	0.359±0.010	3.24±0.07	0.210±0.008
	Transformer	0.697±0.012	0.434±0.019	0.760±0.012	0.419±0.033	8.46±0.07	0.495±0.007	3.95±0.20	0.256±0.016
	LSTM	0.823±0.002	0.608±0.008	0.774±0.002	0.473±0.015	9.16±0.06	0.663±0.008	5.57±0.04	0.681±0.011
(c)	TPC	0.831±0.006	0.645±0.009	0.796±0.006	**0.499±0.016**†	7.24±0.12	0.360±0.005	3.26±0.07	0.210±0.004
	Transformer	0.675±0.052	0.399±0.079	**0.743±0.011**†	0.406±0.022	8.44±0.29	0.492±0.024	3.95±0.25	0.251±0.026
	LSTM	0.820±0.003	0.608±0.003	0.773±0.005	0.473±0.014	9.16±0.04	0.663±0.005	5.60±0.05	0.685±0.010
(d)	TPC	0.832±0.005	0.645±0.016	0.796±0.007	**0.483±0.020**†	7.28±0.09	0.362±0.007	3.29±0.06	0.213±0.002
	Transformer	0.698±0.017	0.431±0.041	**0.743±0.008**†	0.391±0.008	8.44±0.23	0.492±0.019	3.91±0.39	0.253±0.033
	LSTM	0.820±0.003	0.608±0.007	0.773±0.002	0.464±0.011	9.19±0.04	0.669±0.006	5.59±0.03	0.688±0.010
(e)	TPC	0.828±0.004	0.643±0.010	0.798±0.005	**0.480±0.020**†	7.38±0.20	0.367±0.020	3.24±0.07	0.212±0.012
	Transformer	**0.676±0.019**†	0.410±0.034	**0.736±0.021**†	0.383±0.026	8.67±0.27	0.509±0.024	4.12±0.22	0.268±0.017
	LSTM	0.819±0.005	0.604±0.013	0.773±0.002	0.475±0.008	9.20±0.04	0.669±0.008	5.61±0.04	0.691±0.012
(f)	TPC	**0.823±0.006**†	**0.626±0.014**†	**0.793±0.002**†	**0.477±0.017**†	-	-	-	-
	Transformer	0.669±0.036	**0.373±0.048**†	**0.737±0.021**†	0.400±0.038	-	-	-	-
	LSTM	**0.817±0.003**†	**0.597±0.007**†	**0.767±0.003**‡	0.458±0.016	-	-	-	-
(g)	TPC	-	-	-	-	**6.99±0.10**†	**0.341±0.007**†	**3.08±0.09**†	**0.180±0.004**‡
	Transformer	-	-	-	-	**8.18±0.12**‡	**0.472±0.012**†	**3.68±0.18**†	**0.224±0.009**†
	LSTM	-	-	-	-	**9.05±0.05**†	**0.644±0.006**‡	5.55±0.01	**0.668±0.003**†

and compared the resulting clusters to the original (Figure 9). Overall, there is strong cohesion between the models, but sometimes there are shifts in the boundaries between neighbouring clusters in Figure 2. Especially between cluster 5 (moderate-severe) and cluster 1 (severe), where some models (TPC 2, TPC 3, TPC 4) allow patients in cluster 5 to enter their 'cluster 1' equivalent. Nevertheless, very distinct phenotypes are almost never mixed e.g. clusters 2 and 3, 1 and 4, or 1 and 3 (as defined by the TPC 1 model). As in the encoder comparisons, cluster 3 is always well characterised.

Number of Clusters The value of k was determined using the elbow method. In Figure 10, we show how the clusters would appear with increasing value of k. What is most striking is that each time a cluster is added, the new cluster either inserts itself within an existing cluster, or it appears at the intersection between existing clusters. For example, as we move from 2 to 3 clusters, the new cluster 3 is almost completely contained within the old cluster 1. This pattern of sub-dividing an existing cluster generally continues until we reach 6 and 7 clusters, when the new cluster inserts itself at the boundary between two or more old clusters. In other words, increasing the value of k does not completely shift the position of all of the clusters, but rather it carefully subdivides them. The importance of this behaviour with increasing value of k is discussed in the next section.

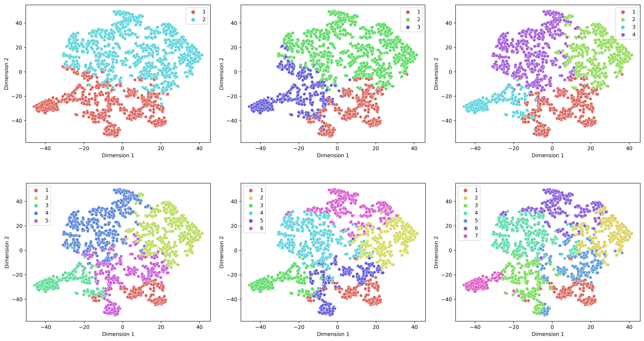

Figure 10: Cluster labels with increasing number of clusters (from k=2 to k=7).

Data Preprocessing

Of the 14,836 episodes extracted from the Amsterdam data, 13,783 ended in extubation or death of the patient, 648 ended with a tracheotomy procedure occurring within 21 days, 399 patients were still on ventilation at 21 days, and 6 patients were converted to a non-invasive ventilation setting. Table 6 shows a summary of the cohort.

Static Features

We extracted 14 static features from the *admissions* table (Table 7). Discrete and continuous variables were scaled to the interval [-1, 1], using the 5th and 95th percentiles as the boundaries, and absolute cut offs were placed at [-4, 4].

Table 6: Cohort summary for the Amsterdam UMC database. 'Remaining LoS' refers to the remaining duration in the hospital after the start of the ventilation episode.

Number of ventilation episodes	14,836
Train	10,395
Validation	2,230
Test	2,208
Sex (% male)	66.6%
Total LoS in days (mean)	8.26
Total LoS in days (median)	2.13
Remaining LoS in days (mean)	7.26
Remaining LoS in days (median)	2.01
VD in days (mean)	3.95
VD in days (median)	0.83
In-hospital mortality	14.6%
Tracheostomy patients	7.4%
'Urgent' patients	28.1%
Number of input features	45
Time series	31
Static	14

Binary variables were coded as 1 and 0. Categorical variables were converted to one-hot encodings, with the exception of 'agegroup', 'heightgroup' and 'weightgroup'. These appear as ordered categories e.g. [18-39, 40-49, 50-59, 60-69, 70-79, 80+] for agegroup. We converted these to an ordered set centred on 0, [-1, -0.6, -0.2, 0.2, 0.6, 1], to preserve the quantitative significance of each category.

Table 7: Static features used in the model. 'Null Height' and 'Null Weight' were added as indicator variables to indicate when the height or weight were missing and have been imputed with the mean value. We added the variables 'Admission Count' and 'Ventilation Episode Count' based on previous admissions and ventilation episodes.

Feature	**Type**	**Source Table**
Sex	Binary	admissions
Age Group	Discrete	admissions
Height Group	Discrete	admissions
Weight Group	Discrete	admissions
Admission Count	Discrete	
Ventilation Episode Count	Discrete	
Urgency	Binary	admissions
Previous Ward	Categorical	admissions
Specific Location in ICU	Categorical	admissions
Physician Speciality	Categorical	admissions
Weight Source	Categorical	admissions
Height Source	Categorical	admissions
Null Height	Binary	
Null Weight	Binary	

Time Series

For each ventilation episode, we selected 31 time series variables, mostly from the *numericitems* table (these are shown in Table 12). We used a semi-automatic process for feature selection. To be included, the variable had to be present in at least 25% of patient stays, and these were further narrowed down with advice from Dr Ari Ercole. We extracted 'diagnosissubgroups' using a query from the AmsterdamUMCdb github repository (Thoral, Driessen, and Dam 2020) and ventilator settings from *listitems*. The ventilator settings classification is given in Table 13. We engineered the features 'lung compliance' and 'P/F ratio' because they are clinically important, and we have previously noted that neural networks are unreliable when performing divisions. We calculated lung compliance as:

$$\text{Lung Compliance} = \frac{0.73556 \times \text{Expiratory Tidal Volume}}{\text{Peak Inspiratory Pressure} - \text{PEEP}} \tag{1}$$

where 0.73556 is a conversion factor to convert lung compliance to its usual unit of ml/cmH$_2$O. We calculated the P/F ratio as:

$$\text{P/F Ratio} = \frac{\text{PaO}_2}{\text{FiO}_2} \tag{2}$$

where FiO$_2$ is expressed as a fraction rather than a percentage.

The time series variables were standardised in the same manner as the static features. To help the model cope with this missing data, we re-sampled according to one-hour intervals and forward-filled the data over the gaps. Note that this is more realistic than interpolation as the clinician would only have the most recent value. After forward-filling was complete, any data recorded before the ICU admission was removed.

Decay Indicators With the forward-filling method alone, the model would not know whether a particular data point was genuine or whether the data had been imputed. This is important because the sampling itself may be informative, for example a deteriorating patient may have more frequent investigations. To mitigate for this, we added 'decay indicators' to specify where the data had been imputed, and if it had, how long it had been since the genuine measurement was taken. The decay was calculated as 0.8^j, where j is the time since the last recording. This is similar in spirit to the masking used by Rocheteau, Liò, and Hyland (2021); Che et al. (2018).

Additional Implementation Details

We tested three different encoder models to generate the embeddings. They were all trained as follows. Firstly, the time series are given to the encoder network which processes and then combines them with the static features. These are then passed through a small two-layer pointwise convolution to generate the embeddings (shown in green on Figure 1). These are given to a predictor network, a reconstruction network and a forecasting network.

The predictor network is one layer, with four outputs, corresponding to the four outcome tasks – tracheostomy, mortality, LoS and VD. For the binary predictions, we apply a sigmoid activation function to generate a prediction between 0 and 1 and for the duration predictions we apply an exponential function. This is intended to help to circumvent a common issue seen in previous models (e.g. Harutyunyan et al. (2019), as they struggle to produce predictions over the full range of durations when the data is very skewed) because it effectively allows the upstream network to model $\log(\text{LoS})$ instead of LoS. The $\log(\text{LoS})$ distribution is much closer to a Gaussian distribution than the remaining LoS. No activation function is placed on the outputs of the forecasting or time series reconstruction networks, because the variables are continuous. Batch normalisation (Ioffe and Szegedy 2015) and dropout (Srivastava et al. 2014) is used to regularise the model.

The LSTM (Hochreiter and Schmidhuber 1997) is very similar to the one used in a recent eICU benchmark paper including LoS prediction (Sheikhalishahi, Balaraman, and Osmani 2019). The Transformer (Vaswani et al. 2017) is very similar to its original implementation except that we added temporal masking to impose causality[5] (see 'Hyperparameter Search Methodology' for their hyperparameters).

TPC Model Architecture

We use a Temporal Pointwise Convolution (TPC) (Rocheteau, Liò, and Hyland 2020) network as one of our encoders. This is a model that takes advantage of both temporal convolution (to analyse trends) and pointwise convolution (to look for any important variable interactions). It is inspired by the way that clinicians would approach an assessment of a patient e.g. they might check how the respiratory rate is changing over time, and they may also look at combination features e.g. the $\text{PaO}_2/\text{FiO}_2$ ratio. The components of the network are briefly explained below.

Temporal Convolution Temporal Convolution Networks (TCNs) (van den Oord et al. 2016; Kalchbrenner et al. 2016) are models that convolve over the time dimension. The TPC model uses stacked TCNs to extract *temporal trends* in the data. Unlike most implementations, it *does not share weights across features* i.e. weight sharing is only across time. This is because the features in a typical EHR differ sufficiently in their temporal characteristics and warrant specialised processing. In TCNs, the receptive field sizes[6] are highly adaptable. They can be increased by using greater dilation, larger kernel sizes or by stacking more layers. By contrast, recurrent neural networks such as LSTM can only process one time step at a time, and Transformers have a weaker sense of temporal structure e.g. periodicity, which is central to understanding time series in the EHR.

Pointwise Convolution Pointwise (or 1x1) convolution (Lin, Chen, and Yan 2013) is typically used to reduce the dimensions in an input (Szegedy et al. 2014). However in the TPC model it is used to compute *interaction features* from the existing feature set at each timepoint.

Skip Connections Skip connections (He et al. 2015) allow each layer to see the original data and the pointwise outputs from previous layers. This helps the network to cope with infrequently sampled data.

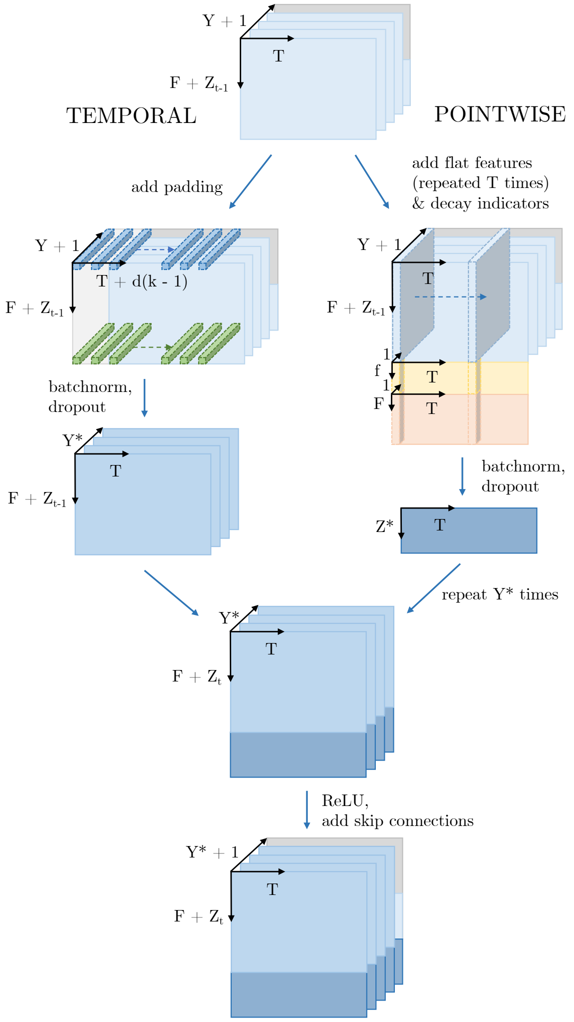

Figure 11: One layer of the TPC model. F is the number of time series features. T is the time series length. Y is the number of temporal channels per feature in the *previous* TPC layer (except for the first layer where Y is 1; decay indicators (explained under 'Time Series' in) make up this channel). Z_{t-1} is the cumulative number of pointwise outputs from *all previous* TPC layers. Y^* and Z^* are the number of temporal channels per feature and pointwise outputs respectively in the *current* TPC layer. $Z_t = Z_{t-1} + Z^*$. The differently coloured temporal filters indicate independent parameters. d is the temporal dilation, k is the kernel size. Decay indicator features () are shown in orange, f static features are shown in yellow. The skip connections consist of F original features (grey) and Z_{t+1} pointwise outputs (light blue). We ignore the batch dimension for clarity.

Temporal Pointwise Convolution The full model combines temporal and pointwise convolution in parallel. Figure 11 shows just one layer, however our implementation has 6 layers stacked sequentially (Table 8). With each successive TPC layer, the temporal dilation is increased by 1.

[5] The processing of each timepoint can only depend on current or earlier positions in the sequence.

[6] 'Receptive field' refers to the width of the filter. For TCNs this corresponds to a timespan.

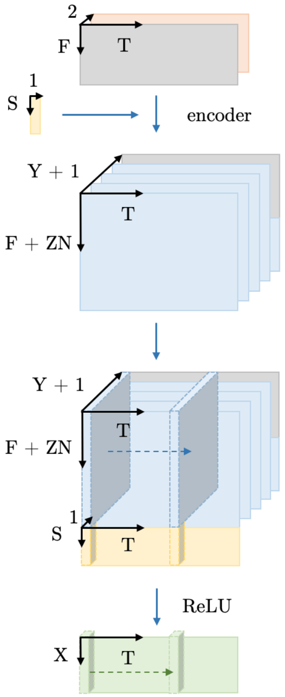

Figure 12: Overview of the encoding framework. F, T, Y*, Z*, Z_t and f are defined in the caption to Figure 11. The original time series (grey) along with the decay indicators (orange) (explained under 'Time Series') are processed by n TPC layers. If a baseline model were used instead of TPC, the time series output dimensions would be M x T, where M is the LSTM hidden size or d_{model} in the Transformer (this is in place of the light blue and grey output in the TPC model). The diagnoses, d, are embedded by a diagnosis encoder – a single fully connected layer of size D. The time series (blue and grey), diagnosis embedding (purple) and static features (yellow) are concatenated along the feature axis, and a two-layer pointwise convolution is applied to obtain the embeddings (green).

Hyperparameter Search Methodology

All the encoders have hyperparameters that can broadly be split into three categories: time series specific, non-time series specific and global parameters (shown in more detail in Tables 8, 9 and 10). The hyperparameter search ranges have been included in Table 11. We ran 10 hyperparameter trials to optimise the remaining parameters for the TPC, LSTM, and Transformer models. The number of epochs was determined by selecting the best validation performance from a model trained over 300 epochs (early stopping was then used for each individual model). All deep learning methods were implemented in PyTorch (Paszke et al. 2019) using PyTorch Lightning (Falcon 2019) and were optimised using Adam (Kingma and Ba 2014).

We also optimised for the weighting between the tasks. We simply multiplied the loss for each component by a hyperparameter. The best overall learning curves were found when the task weighting coefficients were: 0.5 for the duration tasks, 1 for the binary tasks, 0.1 for time series reconstruction and forecasting, and 0.002 for binary feature reconstruction. The reason for the small weighting for binary feature reconstruction was that the task appeared very easy for the models, especially predicting the sex of the patient, and so the representation became dominated with this at the expense of the other tasks.

Number of Clusters

The value of k was determined using an average value from the elbow method across various encoders. Specifically we looked for the point at which the Within Cluster Sum of Squares (WCSS) started to tail off with increasing values of k. Figure 13 shows an example elbow plot. We selected the value 5 across all the models.

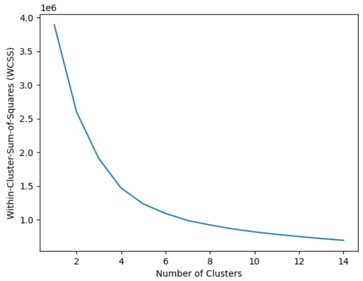

Figure 13: Elbow plot for the TPC model.

Table 8: The TPC model has 11 hyperparameters (Main Dropout and Batch Normalisation have been repeated in the table because they apply to multiple parts of the model). We allowed the model to optimise a custom dropout rate for the temporal convolutions because they have fewer parameters and might need less regularisation than the rest of the model. The best hyperparameter values are shown in brackets. Hyperparameters marked with * were fixed across all of the models.

TPC Specific	
Temporal Specific	**Pointwise Specific**
Temp. Channels (6)	Point. Channels (14)
Temp. Dropout (0.05)	Main Dropout* (0.05)
Kernel Size (3)	
Batch Normalisation* (True)	
No. TPC Layers (6)	
Non-TPC Specific	**Global Parameters**
Batch Normalisation* (True)	Batch Size (128)
Main Dropout* (0.05)	Learning Rate (0.0001)
Final FC Layer Size* (16)	Embedding Size (128)

Table 9: The LSTM model has 8 hyperparameters. We allowed the model to optimise a custom dropout rate for the LSTM layers. Note that batch normalisation is not applicable to the LSTM layers. The best hyperparameter values are shown in brackets. Hyperparameters marked with * were fixed across all of the models.

LSTM Specific	**Non-LSTM Specific**	**Global Parameters**
Hidden State (128)	Batch Normalisation* (True)	Batch Size (128)
LSTM Dropout (0.05)	Main Dropout* (0.05)	Learning Rate (0.0001)
No. LSTM Layers (2)		Embedding Size (128)

Table 10: The Transformer model has 9 hyperparameters. Note that batch normalisation is not applicable to the Transformer layers (the default implementation uses layer normalisation). The best hyperparameter values are shown in brackets. Hyperparameters marked with * were fixed across all of the models.

Transformer Specific	**Non-Transformer Specific**	**Global Parameters**
No. Attention Heads (2)	Batch Normalisation* (True)	Batch Size (128)
Feedforward Size (256)	Main Dropout* (0.05)	Learning Rate (0.0001)
d_{model} (16)		
Transformer Dropout (0.05)		
No. Transformer Layers (6)		

Table 11: Hyperparameter Search Ranges. We took a random sample from each range and converted to an integer if necessary. For the kernel sizes (not shown in the table) the range was dependent on the number of TPC layers selected (because large kernel sizes combined with a large number of layers can have an inappropriately wide range as the dilation factor increases per layer). In general the range of kernel sizes was around 2-5 (but it could be up to 10 for small numbers of TPC Layers).

Hyperparameter	**Lower**	**Upper**	**Scale**
Batch Size	4	512	$\log_2$
Dropout Rate (all)	0	0.5	Linear
Learning Rate	0.0001	0.01	$\log_{10}$
Batch Normalisation	True	False	
Final FC Layer Size	16	64	$\log_2$
Point. Channels	4	16	$\log_2$
Temp. Channels	4	16	$\log_2$
LSTM Hidden State Size	16	256	$\log_2$
d_{model}	16	256	$\log_2$
Feedforward Size	16	256	$\log_2$
No. Attention Heads	2	16	$\log_2$
No. TPC Layers	1	12	Linear
No. LSTM Layers	1	4	Linear
No. Transformer Layers	1	10	Linear

Table 12: Time Series features. The features which do not have a source table were calculated from the other features available in the data. ‘Mandatory Ventilation’ and ‘Patient Triggered’ were calculated from the ventilator settings as outlined in Table 13.

Feature	**Type**	**Source Table**
ABP gemiddeld	Continuous	numericitems
Ademfreq.	Continuous	numericitems
Alb.Chem (bloed)	Continuous	numericitems
Bilirubine (bloed)	Continuous	numericitems
CRP (bloed)	Continuous	numericitems
End tidal CO2 concentratie	Continuous	numericitems
Exp. tidal volume	Continuous	numericitems
Glucose (bloed)	Continuous	numericitems
Hartfrequentie	Continuous	numericitems
Ht (bloed)	Continuous	numericitems
Kalium (bloed)	Continuous	numericitems
Kreatinine (bloed)	Continuous	numericitems
Lactaat (bloed)	Continuous	numericitems
Leuco’s (bloed)	Continuous	numericitems
Natrium (bloed)	Continuous	numericitems
O2 concentratie	Continuous	numericitems
P/F ratio	Continuous	
PC	Continuous	numericitems
PEEP (Set)	Continuous	numericitems
PO2 (bloed)	Continuous	numericitems
Piek druk	Continuous	numericitems
Saturatie (Monitor)	Continuous	numericitems
Temp.	Continuous	numericitems
Thrombo’s (bloed)	Continuous	numericitems
TroponineT (bloed)	Continuous	numericitems
UrineCAD	Continuous	numericitems
lung compliance	Continuous	
mandatory ventilation	Binary	
pCO2 (bloed)	Continuous	numericitems
pH (bloed)	Continuous	numericitems
patient triggered	Binary	
Time in the ICU	Discrete	

Table 13: Ventilator Settings Classification, used to produce the features 'Patient Triggered' and 'Mandatory Ventilation' in Table 12.

Patient Triggered Ventilation	Mandatory Ventilation
Bi Vente	MMV
NAVA	VC
PRVC	PC
PRVC (trig)	Pressure Controled
PS/CPAP (trig)	PC (No trig)
SIMV(PC)+PS	PRVC (No trig)
SIMV(VC)+PS	VC (No trig)
VC (trig)	CPPV
VS	IPPV
SIMV_ASB	SIMV
CPAP	BIPAP
BIPAP-SIMV/ASB	
MMV_ASB	
MMV/ASB	
ASB	
IPPV/ASSIST	
CPPV/ASSIST	
CPPV_Assist	
IPPV_Assist	
SIMV/ASB	
CPAP_ASB	
PS/CPAP	
BIPAP/ASB	
CPAP/ASB	

Table 14: Losses for the reconstruction tasks and forecasting task averaged over 5 independent training runs. The error margins are 95% confidence intervals. See 'Reconstruction and Forecasting' for explanations of the losses shown. The meaning of (a), (b), the colour scheme and statistical tests are defined in the legend to Table 1.

	Model	Reconstruction Tasks			Forecasting
		Last Timestep	Static (Binary)	Static (Other)	
(a)	TPC	0.334±0.004	0.013±0.000	0.210±0.038	0.334±0.005
	Transformer	0.351±0.005	0.013±0.000	0.354±0.005	0.347±0.001
	LSTM	**0.297±0.006**‡	**0.012±0.001**†	**0.078±0.010**‡	**0.299±0.004**‡
(b)	TPC	**0.345±0.002**‡	0.013±0.000	**0.332±0.006**‡	**0.345±0.003**‡
	Transformer	0.355±0.006	**0.013±0.000**†	0.356±0.001	**0.353±0.005**†
	LSTM	**0.322±0.003**‡	0.012±0.000	**0.266±0.004**‡	**0.323±0.003**‡

Table 15: Reconstruction and forecasting losses for the task ablation study. The full task setting from Table 14(a) has been repeated for ease of comparison. The following task ablations are compared to (a): (c) includes all tasks except for the forecasting task, (d) includes all tasks except for the reconstruction tasks. The colour scheme and statistical test comparisons are explained in the legend to Table 1.

	Model	Reconstruction Tasks			Forecasting
		Last Timestep	Static (Binary)	Static (Other)	
(a)	TPC	0.334±0.004	0.013±0.000	0.210±0.038	0.334±0.005
	Transformer	0.351±0.005	0.013±0.000	0.354±0.005	0.347±0.001
	LSTM	0.297±0.006	0.012±0.001	0.078±0.010	0.299±0.004
(c)	TPC	0.334±0.004	0.012±0.000	0.198±0.020	-
	Transformer	0.349±0.007	0.013±0.000	0.358±0.007	-
	LSTM	**0.305±0.003**†	0.011±0.000	0.085±0.010	-
(d)	TPC	-	-	-	0.339±0.006
	Transformer	-	-	-	**0.355±0.007**†
	LSTM	-	-	-	**0.309±0.006**†