CLINICALLY RELEVANT UNSUPERVISED ONLINE REPRESENTATION LEARNING OF ICU WAVEFORMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Univariate high-frequency time series are dominant data sources for many medical, economic and environmental applications. In many of these domains, the time series are tied to real-time changes in state. In the intensive care unit, for example, changes and intracranial pressure waveforms can indicate whether a patient is developing decreased blood perfusion to the brain during a stroke, for example. However, most representation learning to resolve states is conducted in an offline, batch-dependent manner. In high frequency time-series, high intra-state and inter-sample variability makes offline, batch-dependent learning a relatively difficult task. Hence, we propose Spatial Resolved Temporal Networks (SpaRTeN), a novel composite deep learning model for online, unsupervised representation learning through a spatially constrained latent space. SpaRTeN maps waveforms to states, and learns time-dependent representations of each state. Our key contribution is that we generate clinically relevant representations of each state for intracranial pressure waveforms.

1 INTRODUCTION

In high-frequency time series data like intracranial pressure waveforms, rapidly predicting and detecting changes in state is a clinically important task. For example, if a patient in the intensive care unit starts exhibiting intracranial pressure decompensation, it may cause bilateral blindness (Mollan et al. (2016)). Consequently, early detection of state transitions may provide clinicians with the tools to intervene appropriately for better outcomes. For example, at early stages, cerebral vascular decompensation can be treated with a loop diuretic like furosemide (Llwyd et al. (2022)). Second, a growing amount of research is indicating the need to redefine critical illness by biological state rather than a non-specific illness syndrome (Maslove et al. (2022)). SpaRTEn takes a key step towards individualized state definition by generating individualized state representations.

We propose a composite differentiable unsupervised deep learning network to learn a discrete spatial representation from a high frequency time series via temporal ensemble learning–a method called **Spa**tial **R**esolved **Te**mporal **N**etworks (SpaRTeN). While we demonstrate one variant of this network, the overall framework can be generalized to many use cases, different network blocks and optimization procedures. We train the network via back-propagation and generate spatial representations via forward propagation.

1.1 CONTRIBUTIONS

- We introduce SpaRTEn, a new framework for learning spatial representations from high-frequency time series.
- We demonstrate that introduction of a latent space improves rather than harms SpaRTEn's ability to forecast and cluster high frequency data in real-time.
- We show that SpaRTeN can generate clinically meaningful representations of intracranial pressure waveforms.

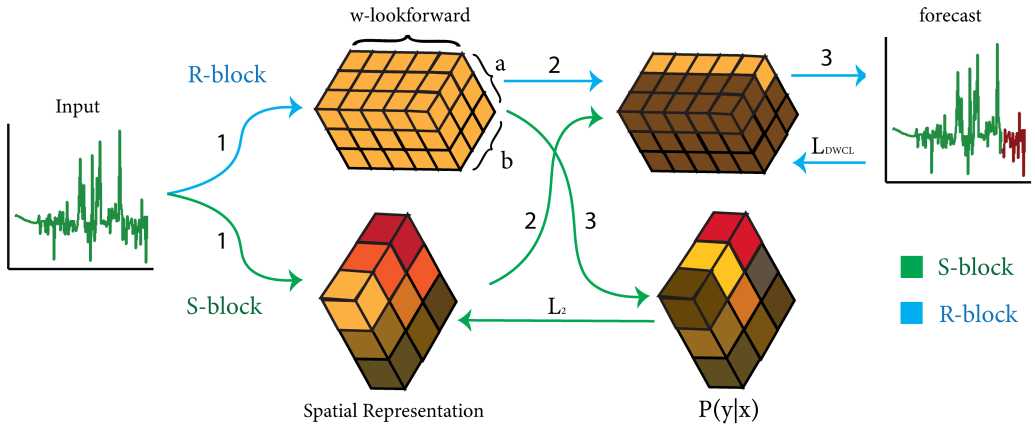


Figure 1: **Schematic representation of model architecture.** Blue is forward propagation and green is back-propagation. Time series from the data space are forecast by the sub-networks in an R-block with a look-forward period of k. The dimensionality of the spatial representation is a $a \times b$. Simultaneously, an S-Block predicts the most relevant sub-network for the time series. The predicted sub-network is used to generate a forecast and is then back-propagated via Equation 9.

2 Background

2.1 Problem Setup.

For reference, we include the mathematical notation in Appendix A.1.

We consider a function mapping a time series $\boldsymbol{x_t}$ of length k, $\{x_{t-k}, x_{t-k+1}, \ldots, x_t\}$ to a state $\boldsymbol{s_t}$. Furthermore, $\boldsymbol{s_t}$ is constrained to represent a discrete point in the positive space of integers $\mathbb{Z}^{2+}$.

$$\boldsymbol{s_t} = (\boldsymbol{x_t}; \theta) \tag{1}$$

$$\boldsymbol{s_t} = \{(i, j); i : [0, a), j : [0, b); i, j \in \mathbb{Z}^+ \tag{2}$$

where i and j respectively represent the x- and y-coordinates of $\boldsymbol{s_t}$. i and j are constrained to be positive integers less than the width a and height b of the latent space. This problem setup imposes a spatial constraint on associative memories learned by Hopfield networks Ramsauer et al. (2020). Self-organizing feature maps (SOMs) also map inputs onto two dimensional Euclidean spaces, but are offline and constrained to inputs of fixed dimension Kohonen (1990). We choose to place $\boldsymbol{s_t}$ in a two-dimensional discrete state-space (i, j), because it facilitates easy visualization of time series corresponding to individual states, which previous methods like SOM-VAE and TFT are unable to currently do.

3 Model Framework

3.1 Composite Neural Network

A schematic overview of our composite model is depicted in Figure 1. The architecture we construct consists of a two-block model with an *S* block and an *R* block. The first block, *S*, generates a mapping from a time series $\boldsymbol{x_t}$ to a state $\boldsymbol{s_t}$. The second block, *R*, includes a spatially constrained ensemble of sub-networks $R_{(.)}$ each of which corresponds to a state and makes a forecast $\hat{\boldsymbol{y}}_{\boldsymbol{s_t}}$ with an arbitrary forecast window w, $\{x_{t+1}, \ldots, x_{t+w}\}$. We train *S* to maximize the probability of assigning the time series to the most suitable sub-network in *R*, which corresponds to the minmax framework proposed in Goodfellow et al. (2014) (Equation 5) .

$$f_S : \boldsymbol{x_t} \rightarrow \boldsymbol{s_t} \tag{3}$$

$$f_{R_{s_t}} : \boldsymbol{x_t} \rightarrow \hat{\boldsymbol{y}}_{\boldsymbol{s_t}} \tag{4}$$

$$\min_R \max_S E_{t \in \tau} L(R(S(\boldsymbol{x_t})) \tag{5}$$

where L is an objective function, R is the ensemble of state networks, S is a function that maps a time series to a density across the states, and t is drawn from the set of all time series τ.

Sub-networks can have a primary objective ($L_{\text{objective}}$) of forecasting, classification, or reconstruction if we assign $\boldsymbol{y_t}$ to be the forecasting window, a labeled class or $\boldsymbol{x_t}$, respectively. However, to further impose structure, we introduce a secondary loss for time series inspired by contrastive loss and self-organizing maps, the distance-weighted contrastive loss (L_{DWCL}). The secondary objective ensures that a state learned by a sub-network in the R-block is unique. This provides a distinct advantage over previous methods that can embed similar states with different embeddings — states with unique embeddings improve interpretation (A.2), which we show via ablation studies (A.6). For a single input, we define the loss function for to be

$$L_{R_{s_t}} = L_{\text{objective}}(\boldsymbol{y_t}, R_{\boldsymbol{s_t}}(\boldsymbol{x_t}; \theta_{R_{s_t}})) + \alpha \times L_{\text{DWCL}}(R_{\boldsymbol{s_t}}(\boldsymbol{x_t}; \theta_{R_{s_t}}), R(\boldsymbol{x_t}; \theta_R)) \quad (6)$$

$$L_{\text{DWCL}}(R_{\boldsymbol{s_t}}, R) = -\log \frac{e^{R_{s_t}}}{E_{\boldsymbol{z}\sim Z}\left[e^{\text{sim}(R_{\boldsymbol{z}}, R_{\boldsymbol{s_t}})} \times ||\boldsymbol{s_t} - \boldsymbol{z}||_2\right]} \quad (7)$$

$$L_s = \|S(\boldsymbol{x_t}) - L(R(\boldsymbol{x_t}))\|_2 \quad (8)$$

where α is a learned hyper-parameter to modulate the relative effects of the two losses. For the examples in this paper, we consider the primary objective to be forecasting. The S block has a separate loss function L_s, contingent on its objective, which is to predict the spatial state occupied by the next time step. S outputs a density function across the state-space domain, and the target for the S-block is the best sub-network in the R-block. While an L_1 or L_2 norm may better capture the information about the density of the spatial networks, we can improve stability by treating the objective as a classification problem and minimize the negative log likelihood (A.4). Training details are further discussed in (Figure 1, A.7).

4 Application

For our applications, we focus on three distinct tasks involving high frequency time series — online forecasting, zero-shot clustering and clinically significant representation learning. First, we benchmark on standard time series data, using SOTA approaches on standard datasets. Second, we apply these results to intracranial pressure waveforms. We benchmark against SOTA online forecasting models with convolutional approaches such as N-Beats (Oreshkin et al. (2019b)) and attention based methods like Temporal Fusion Transformers (Lim et al. (2019)) and Autoformers (Wu et al. (2021)), where SpaRTEn outperforms on three out of the four datasets (Table 1). Zero-shot clustering is an unsupervised method that involves classifying the time series the first time it is seen without information corresponding to a label. We demonstrate that SpaRTeN can generate prototypical waveforms, that can be utilized by K-Nearest Neighbors to perform state-of-the-art for zero-shot clustering methods. We benchmark on traditional clustering techniques such as KNN with random subsets (Rand), Gaussian Mixture Modeling (GMM) and Spectral Clustering (Spec). Ablation studies show that L_{DWCL}, the *S*, and appropriate hyper-parameter selection are crucial to learning meaningful representations (A.6).

Table 1: Benchmarking SpaRTEn against Online Forecasting and Clustering

	Models (RMSE)					Models (Silhouette)			
Dataset	LSTM	N-Beats	TFT	Auto	SRTN	Rand	Spec	GMM	SRTN
Electricity	2.93	2.84	2.49	6.61	**1.57**	0.023	0.005	0.024	**0.028**
Traffic	32.10	3.1	15.1	3.34	**1.58**	0.22	0.09	0.12	**0.24**
Stock	0.13	**0.10**	0.11	1.24	0.67	0.012	0.005	0.011	**0.026**
Retail	13.76	14.16	13.9	4.98	**1.59**	0.011	0.002	0.010	**0.027**

4.1 Time is brain: Clinically Relevant Representation Learning

We demonstrate the practical application of SpaRTEn on intracranial pressure waveforms from the MIMIC-IV data-set. Intracranial pressure waveforms are a good example for this assessment because identifying underlying states may provide insight into an appropriate treatment regimen to

avoid an acute neurological injury (Desai et al. (2019)). A good interpretable representation of states should be a) able to demonstrate distinct properties within each state, and b) cluster waveforms within a given time series into a given state. We highlight SpaRTEn's visualized states and demonstrate the relevant clinical vignettes for each state (Figure 2d).

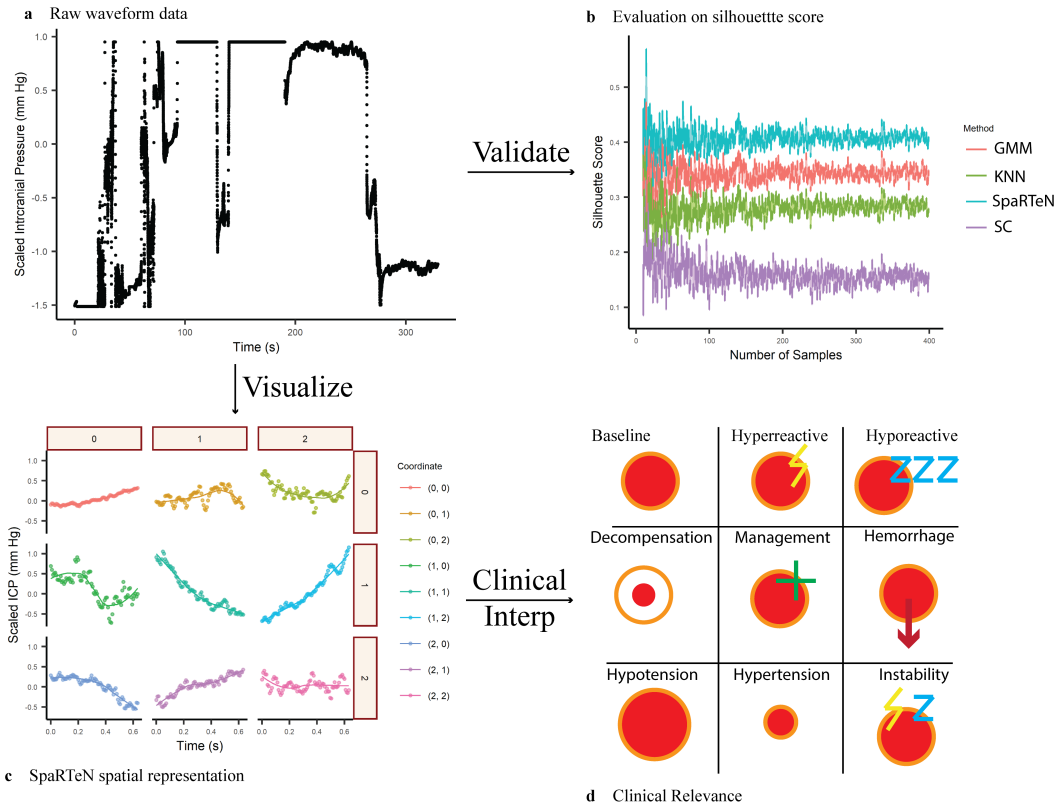


Figure 2: **Qualitatively and quantitatively evaluating model representations.** a) Raw waveforms of Intra-cranial Pressure with high intra-sample variability. b) Bootstrapped (10x) results of silhouette score across different sized samples demonstrates that SpaRTeN outperforms other clustering methods. 95% Confidence intervals reported in the figure, but may be too small to see. c) Clusters generated by SpaRTeN represent distinct trends within the time series. d) Clinical interpretation of each of the waveforms.

At point (0,0), the waveform is both stable and relatively constant, providing a strong baseline for what non-pathological activity should look like. At point $(0, 1)$ we start seeing evidence of pathological neuro-vascular activity - the mean change in the intracranial pressure is relatively small, unlike the variation, which is relatively large. ICP variability is part of the response to injuries like trauma (Svedung Wettervik et al. (2020)). Following trauma, high intracranial pressure variability represents a hyper-active vasogenic, regulatory response. In contrast, at point $(0, 2)$, intracranial pressure waveform has a U-shaped, which indicates a slower, hypo-reactive compensatory response to changes in intracranial pressure. A hypo-reactive intracranial pressure is associated with worse performance on the Glasgow Coma Scale (Tian et al. (2013)). A physician may try to shift a patient's state from $(0, 2)$ to $(0, 1)$ in order to improve outcomes by increasing the tone of the sympathetic nervous system (Schmidt et al. (2018)). SpaRTEn's other representations clinically correlate to hemorrhage, decompensation and hypertension (A.8)

SpaRTeN representations quantitatively capture variations within the data, and qualitatively provide key clinical insights into waveform patterns. Future work should seek to include multi-modal time series and improve size, algorithm and optimization details (A.9).

5 REPRODUCIBILITY STATEMENT

All experiments were performed with PyTorch. The code for the algorithm is attached in the supplementary material.

6 ETHICS STATEMENT

Experiments with publicly available de-identified data from the MIMIC-III Waveform Database were conducted with IRB approval. All other datasets that we used are also publicly available. This work is not expected to lead to negative societal implications.

REFERENCES

Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. *arXiv preprint arXiv:2006.05205*, 2020.

Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

J Alfredo Caceres and Joshua N Goldstein. Intracranial hemorrhage. *Emergency medicine clinics of North America*, 30(3):771–794, 2012.

Shashvat M Desai, Marcelo Rocha, Tudor G Jovin, and Ashutosh P Jadhav. High variability in neuronal loss: time is brain, requantified. *Stroke*, 50(1):34–37, 2019.

Vincent Fortuin, Matthias Hüser, Francesco Locatello, Heiko Strathmann, and Gunnar Rätsch. SOM-VAE: Interpretable discrete representation learning on time series. June 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z Ghahramani, M Welling, C Cortes, N Lawrence, and K Q Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

R Hadsell, S Chopra, and Y LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 1735–1742, June 2006.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. November 2019.

T Kohonen. The self-organizing map. *Proc. IEEE*, 78(9):1464–1480, September 1990.

Andreas H Kramer. Critical icp in subarachnoid hemorrhage: how high and how long? *Neurocritical Care*, 34(3):714–716, 2021.

Bryan Lim, Sercan O. Arik, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting, 2019. URL `https://arxiv.org/abs/1912.09363`.

Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *Int. J. Forecast.*, 37(4):1748–1764, October 2021.

Osian Llwyd, Jui-Lin Fan, and Martin Müller. Effect of drug interventions on cerebral hemodynamics in ischemic stroke patients. *Journal of Cerebral Blood Flow & Metabolism*, 42(3):471–485, 2022.

Niklas Luetzen, Philippe Dovi-Akue, Christian Fung, Juergen Beck, and Horst Urbach. Spontaneous intracranial hypotension: diagnostic and therapeutic workup. *Neuroradiology*, 63(11):1765–1772, 2021.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The M4 competition: Results, findings, conclusion and way forward. *Int. J. Forecast.*, 34(4):802–808, October 2018.

David M Maslove, Benjamin Tang, Manu Shankar-Hari, Patrick R Lawler, Derek C Angus, J Kenneth Baillie, Rebecca M Baron, Michael Bauer, Timothy G Buchman, Carolyn S Calfee, et al. Redefining critical illness. *Nature Medicine*, 28(6):1141–1148, 2022.

Susan P Mollan, Fizzah Ali, Ghaniah Hassan-Smith, Hannah Botfield, Deborah I Friedman, and Alexandra J Sinclair. Evolving evidence in adult idiopathic intracranial hypertension: pathophysiology and management. *Journal of Neurology, Neurosurgery & Psychiatry*, 87(9):982–992, 2016.

Sanath Narayan, Akshita Gupta, Salman Khan, Fahad Shahbaz Khan, Ling Shao, and Mubarak Shah. Discriminative region-based Multi-Label Zero-Shot learning. August 2021.

Eva K Oernbo, Annette B Steffensen, Pooya Razzaghi Khamesi, Trine L Toft-Bertelsen, Dagne Barbuskaite, Frederik Vilhardt, Niklas J Gerkau, Katerina Tritsaris, Anja H Simonsen, Sara D Lolansen, et al. Membrane transporters control cerebrospinal fluid formation independently of conventional osmosis to modulate intracranial pressure. *Fluids and Barriers of the CNS*, 19(1): 1–25, 2022.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. May 2019a.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting, 2019b. URL `https://arxiv.org/abs/1905.10437`.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield networks is all you need. July 2020.

Pantelis Samartsidis, Shaun R Seaman, Anne M Presanis, Matthew Hickman, and Daniela De Angelis. Assessing the causal effect of binary interventions from observational panel data with few treated units. April 2018.

Eric A Schmidt, Fabien Despas, Anne Pavy-Le Traon, Zofia Czosnyka, John D Pickard, Kamal Rahmouni, Atul Pathak, and Jean M Senard. Intracranial pressure is a determinant of sympathetic activity. *Frontiers in physiology*, 9:11, 2018.

Teodor Svedung Wettervik, Timothy Howells, Per Enblad, and Anders Lewén. Intracranial pressure variability: relation to clinical outcome, intracranial pressure–volume index, cerebrovascular reactivity and blood pressure variability. *Journal of clinical monitoring and computing*, 34(4): 733–741, 2020.

Ye Tian, Zengguang Wang, Ying Jia, Shengjie Li, Bin Wang, Shizhao Wang, Lin Sun, Jianning Zhang, Jieli Chen, and Rongcai Jiang. Intracranial pressure variability predicts short-term outcome after intracerebral hemorrhage: a retrospective study. *Journal of the neurological sciences*, 330(1-2):38–44, 2013.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proc. IEEE*, 109(1):43–76, January 2021.

A APPENDIX

A.1 MATHEMATICAL NOTATION

Table 2: Mathematical Notation

Symbol	Meaning
$\boldsymbol{x_t}$	A time series from $t-k$ to t. $\{x_{t-k}, x_{t-k+1}, \ldots, x_t\}$. k is the look-back period.
$\boldsymbol{y_t}$	A time series from $t+1$ to $t+w$. $\{x_{t+1}, \ldots x_{t+w}\}$. w is the forecast window.
$\boldsymbol{s_t}$	The state of a time series at time-point t.
	Corresponds to (i, j), a coordinate within the latent space.
	Within S, the range of possible states
$\boldsymbol{\hat{y}_{t_s}}$	Prediction of y_t given state s
T	Total length of time series.
i	The x-coordinate of a state. Constraint: $i < a$
j	The y-coordinate of a state. Constraint: $i < b$
Z	The latent space of states. Constrained to $\{\mathbb{Z}^{2+} : [0, a), [0, b)\}$
a	The width of the latent space.
b	The height of the latent space.
f_S	$f_S : X_t \rightarrow s_t$. The S-block.
f_R	$f_R : (s, X_t) \rightarrow y_t; \forall s \in S$. The R-Block.
R_s	A network in the R-block. Maps $X_t \rightarrow \hat{y}_{t\,s}$ given s
θ_f	Parameter of function f.
L	Loss function.
sim	A metric of similarity. Normalized dot produce or cosine similarity, for example.

A.2 DISTANCE-WEIGHTED CONTRASTIVE LOSS

We adapt contrastive loss and self-organizing maps in the second term to the distance-weighted contrastive loss (DWCL). For a single sample,

$$L_{\text{DWCL}}(R_{\boldsymbol{s_t}}, R) = -\log \frac{e^{R_{\boldsymbol{s_t}}}}{E_{\boldsymbol{z}\sim Z}\left[e^{\text{sim}(R_{\boldsymbol{z}}, R_{\boldsymbol{s_t}})} \times ||\boldsymbol{s_t} - \boldsymbol{z}||_2\right]} \tag{9}$$

where $\boldsymbol{z}$ is a state drawn from the set of all states Z, sim is a metric of similarity between $R_{\boldsymbol{z}}$ and $R_{\boldsymbol{s_t}}$, such as a normalized dot product or cosine-similarity.

This loss pushes sub-networks to have distinct predictions. The distance-weighted contrastive loss for univariate time series learns similar and dissimilar pairs in a self-supervised manner. During each forward propagation step, the predictions are generated by each forecasting network in the R-block. After the predictions are generated by each network in the R block, the distance-weighted contrastive loss is created by calculating the difference in the predictions between each of the network in the R-block and network selected by the S block, and weighting it by Euclidean distance of the network and the selected sub-network. Similar pairs can be thought of networks that are closer to the selected network in euclidean space, and dissimilar pairs are those that are further apart in Euclidean space.

The overall computational cost is $O(c(f, b) + (K-1) \times c(f))$, where $c(f)$ is the cost of forward propagating, and $c(f, b)$ is the cost of forward and back-propagating, and K is the total number of blocks. Thus, computational cost scales with the number of sub-networks.

The general idea of contrastive loss is to preserve neighborhood relationships between data points by minimizing the distance between similar points and maximizing the distance between points of different classes Hadsell et al. (2006). The general form of the contrastive loss function is the following:

$$L_{\text{contrastive}}(x_i, x_j; \theta) = \frac{1[y_i = y_j]}{2} d(f_\theta(x_i), f_\theta(_j)) + \frac{1[y_i \neq y_j]}{2} \max(0, \epsilon - d(f_\theta(x_i), f_\theta(x_j)) \tag{10}$$

where x_i and x_j are two distinct samples, f is a function that maps $x \rightarrow R^k$, an embedding of dimensionality k, d is a distance metric, and ϵ is the distance to the margin. In multi-class classification problems, this can be further extended as a classification problem with $K+1$ categories He et al. (2019).

$$L_q = -\log \frac{e^{\text{sim}(f_\theta(x_t), f_\theta(x_j))/\tau}}{\sum_{k=1}^{N} e^{\text{sim}(f_\theta(x_k), f_\theta(x_j))/\tau}} \tag{11}$$

where $\text{sim}(f(x_i), f(x_j))$ is a metric of similarity between $f(x_i)$ and $f(x_j)$ and τ is the normalization factor.

We can extend this to forecasting where the positive example can be thought of as the selected sub-network, whereas the negative examples are the irrelevant sub-networks. Finally, we add a normalized distance metric, to ensure sub-networks that are closer in euclidean space have closer representations.

$$L_{\text{DWCL}}(R_{i,j}, R) = -\log \frac{e^{R_{i,j}}}{E_{x,y \sim \mathbb{Z}^{2+}} \left[e^{\text{sim}(R_{x,y}, R_{i,j})} \times \frac{\sqrt{(x-i)^2+(y-j)^2}}{\sum_{x,y}^{\mathbb{Z}^{2+}} (x-i)^2+(y-j)^2} \right]}$$

We visualize this further in Figure 3.

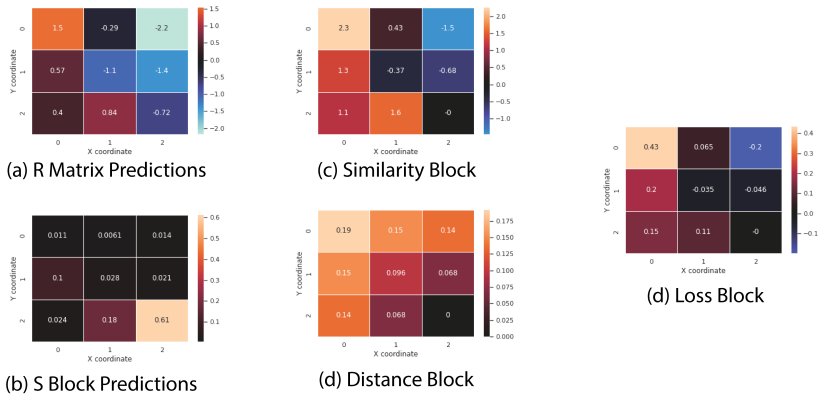


Figure 3: Loss calculations for the R-matrix. (a) Is the calculation of the R-matrix predictions for the next time step. (b) S-Block predicts the appropriate state for the next time step. (c) The similarity block calculates similarity between the chosen state prediction and the other states. (d) The distance block calculates the distance between each other state and the selected substates. The loss block is the dot product of the distance and similarity block. The loss block is summed to produce the final loss value

A.3 Inductive Biases with Ensemble Weight Sharing

Inductive transfer learning leverages an inductive bias to improve performance on a target task and eliminates redundant learning of patterns in data structure Zhuang et al. (2021). To generate sub-networks with weights that represent distinct states rather than shared structure between states in the time series, we employ an inductive transfer learning framework. This procedure increases the gap between the sub-network posteriors, which further enhances the contrastive learning aspect of the network. From an information theoretical perspective, the process of learning a shared posterior can be thought of as a lossless compression of the hidden states by encoding them into a shared embedding. In turn, non-unique learning of state-independent behavior only needs to take place once rather than $a \times b$ times.

While mode collapse is a known problem, we find that the sharing of weights across the first few layers leads to robust performance as seen with the paradigm of transfer learning. This is quite unlike the mode collapse seen in the training of generative adversarial networks. Without the sharing of weights across the first few layers, we find that learning requires significantly more samples because

the overall structure of the time series must be learned for each unit in the R block, in addition to learning of the relevant state. In contrast, with weight sharing, we find that learning of the overall structure of time series can be done jointly, and the state separation can be learned by each sub-network.

In our implementation of the SpaRTeN framework, this auxiliary network maps the hidden states of R, which is an $h \times a \times b$ embedding to a low-dimensional embedding h', which is subsequently appended to a dense layer of each sub-network. This procedure ensures that the shared weights are differentiable during training. After back-propagation, weights are copied to all sub-networks. The full training algorithm is provided in Algorithm 1.

Algorithm 1 Spatial Projection of Time Series with Temporal Ensembles

Require: $\{x_0, \ldots, x_T\}$, where T is the length of the time series. $\boldsymbol{x_t}$ represents $\{x_{t-k}, \ldots, x_t\}$ where k is the look-back period, and y_t is the forecast period $\{x_{t+1}, \ldots, x_{t+w}\}$ where w is the forecast window. Assign a width and height to the latent space $a, b \in \mathbb{Z}^+$. Randomly initialize weights of the R and S block: $\theta_R, \theta_S \sim N(0, 1)$.

for $m = k$ to $m = T - w$ **do**

$\boldsymbol{x_m} \leftarrow \{x_{m-k}, \ldots, x_m\}$

$\boldsymbol{y_m} \leftarrow \{x_{m+1}, \ldots, x_{m+w}\}$

$(\hat{i, j}) = S(\boldsymbol{x_m}; \theta_S)$

$\hat{\boldsymbol{y}}_{\boldsymbol{m}(i,j)} = R_{(i,j)}(\boldsymbol{x_m}; \theta_R) \forall \{i : [0, a), j : [0, b)\}$

Update $R_{(\hat{i,j})}$ via gradient descent on $L_{DWCL}(\hat{\boldsymbol{y}}_{\boldsymbol{m}(\hat{i,j})}, \boldsymbol{y_m})$

Update S via gradient descent on $L(S(\boldsymbol{x_m}), \arg\min_{i,j}(L(\hat{\boldsymbol{y}}_{\boldsymbol{m}(i,j)}, \boldsymbol{y_m}))$

end for

A.4 Encouraging smoothness over time

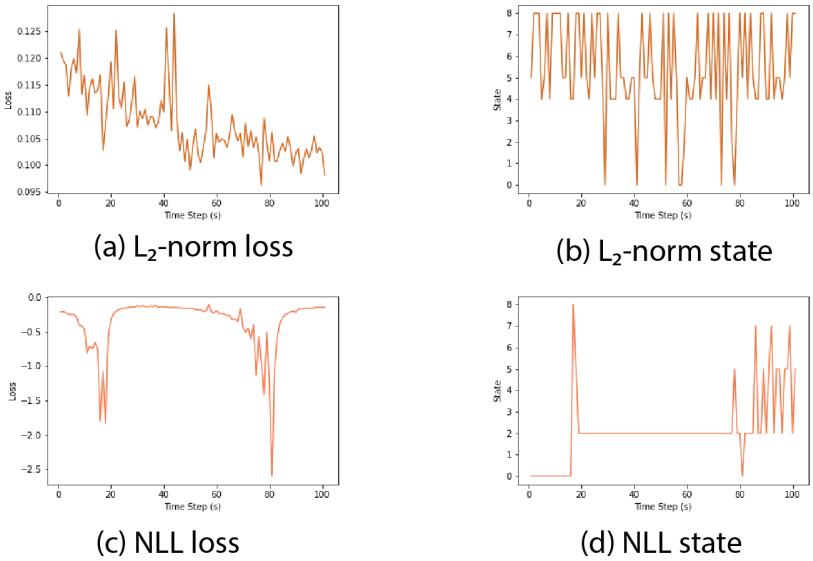
The goal is to predict the development of a time series in an interpretable way. This means that we may have a tradeoff between stable network dynamics and representation of a ground truth density. Learning a probabilistic model in a high-dimensional continuous space can be challenging, which necessitates the use of reductionist frameworks to improve interpretability.

Previous work in Markov chain modeling penalized state transitions via an additional smoothness term Fortuin et al. (2018). Other methods have focused on incorporating quantile outputs to maximize the signal-to-noise ratio Lim et al. (2021).

We find that by converting an L_2-norm-based loss function to cross-entropy loss, we can improve the stability of both the S-block representations, and by extension, the R-block ensemble:

$$L_S = -\sum_{i,j}^{\mathbb{Z}^{2+}:[a,b]} \arg\min_{i,j}(R_{i,j}(\boldsymbol{x_t}) - x_{t+1})^2 \times \log \sigma(S(\boldsymbol{x_t})) \tag{12}$$

where $\mathbb{Z}^{2+}$ is a discrete two-dimensional space of integers in $[a, b]$, the sum is over all the coordinates in the space, $R_{i,j}(\boldsymbol{x_t})$ is the prediction of the next time step by the network based on the previous time step, x_{t+1} is the next step. σ represents soft-max function, and $S(\boldsymbol{x_t})$ is the predicted state of the next time step. If S fails to provide strong initial gradients, as in the case with L_2-norm, then the instability of the network prevents a single sub-network from learning the characteristics of a given state (Figure 4). In turn, this causes the S-block to be increasingly volatile, which can in turn further destabilize the R-block.



(a) L_2-norm loss (b) L_2-norm state

(c) NLL loss (d) NLL state

Figure 4: Losses and associated states for L_2-norm and negative log-likelihood. (a) L_2-norm loss is significantly smaller and signal-to-noise ratio is smaller than (c) negative log likelihood (NLL) loss. The corresponding states calculated by the (b) L_2 loss are far more unstable than the states calculated by the (d) negative log likelihood S-block.

A.5 Benchmarks

We benchmark against the UCI electricity, UCI traffic dataset, the five-min sub-sampled realized volatility from the Oxford stocks dataset, and the kaggle retail dataset Asuncion & Newman (2007). We benchmark on long short-term memory networks (LSTMs), N-Beats, and temporal fusion transformers on an online forecasting task. We report root mean squared error (RMSE) Oreshkin et al. (2019a); Lim et al. (2021).

Table 3: RMSE of benchmarks on an online forecasting task

	Datasets			
Model	Electricity	Traffic	Stocks	Retail
LSTM	2.93	32.10	0.13	13.76
N-Beats	2.84	3.10	**0.10**	14.16
TFT	2.49	15.10	0.11	13.87
Autoformer	6.61	3.34	1.24	4.98
SpaRTEn	**1.57**	**1.58**	0.67	**1.59**

Second, we benchmark the representations learned via SpaRTEn via a clustering metric, silhouette score. We benchmark on the same datasets above, and compare against standard models such as k-means applied to random sampling as a baseline; spectral clustering, which imposed a graph-based approach to clustering and is typically used for sequential genomic data; and, a Gaussian mixture model. We show that the silhouette score for the SpaRTeN representations far exceeds other modes of clustering.

Table 4: Silhouette score of benchmarks on an unsupervised clustering task

	Datasets			
Model	Electricity	Traffic	Stocks	Retail
Random	0.023	0.22	0.012	0.011
Spectral	0.005	0.09	0.005	0.002
GMM	0.024	0.12	0.011	0.010
SpaRTEn	**0.028**	**0.24**	**0.026**	**0.027**

A.6 Ablation Studies

In this section, we run four different ablation studies. The baseline model contains a latent space of dimension 3×3, a negative log-likelihood loss, an S-Block and an R-Block. We perform three distinct experiments.

First, we ablate the S-Block. Ablation of the S-Block significantly decreases the performance of the model. We anticipate this is because the spread of patterns included in the time series analysis are subject to oversquashing Alon & Yahav (2020).

Second, we over-parameterize the latent space to a 10×10. We show that this slightly decreases the performance, but not by much in the online forecasting task across three of the four datasets. Because there are no constraints on how many coordinates the network needs to use, this may simply be the result of self-regularization where the network voluntarily learns a representation that under-utilizes an over-parameterized space. Nevertheless, over-specification of the latent space harms the clustering ability of SpaRTEn.

Third, we ablate the distance-weighted contrastive loss. The distance-weighted contrastive loss was implemented to improve clustering. We see that eliminating the distance-weighted contrastive loss can reduce online forecasting performance and clustering performance.

Table 5: Ablation study on benchmarked datasets

	Dataset							
	UCI Electricity		UCI Traffic		Oxford Stocks		Retail	
Ablation	RMSE	Silhouette	RMSE	Silhouette	RMSE	Silhouette	RMSE	Silhouette
S-block	2.93	**X**	32.10	**X**	**0.13**	**X**	13.76	**X**
10×10	2.58	0.019	**1.58**	0.15	0.57	0.005	1.61	0.001
DWCL	2.57	0.024	1.62	0.17	0.60	0.012	1.60	0.013
None	**1.58**	**0.028**	**1.58**	**0.24**	0.67	**0.026**	**1.59**	**0.027**

A.7 Model Architecture and Training

We employ a model architecture that utilizes the SpaRTeN framework. It consists of an S block and an R block with a spatial embedding space of $\mathbb{Z}^{2+} : [0, a), [0, b)$ (Figure 1).

Each step in training occurs in three progressive steps during forward propagation and two steps during back-propagation (Figure 1). During forward propagation, the R block maps an input time series to a set of forecasts $a \times b$ (Step 1 - Blue). The S block takes the input time series and generates a spatial density over the states (Step 1 - Green).

The second step is that the spatial densities corresponding to the predicted state over the S block is used to select the network in the R block to predict the time steps over the look-forward period (Step 2 - Blue, Green).

During backpropagation, there are two distinct loss functions that must be accounted for. First, the S block loss can be calculated by generating all the predictions in networks in the R blocks (Step 3 - Green). The difference between S-block prediction of the best state and the network in the R block with the lowest error with respect to the true future values (in the case of online forecasting),

can be calculated (L_S), and the subsequently it can be either treated as a classification task with cross-entropy loss or a mean-squared error loss with the S block trying to approximate the density generated by the R networks.

Second, the R block loss can be calculated by utilizing the predictions generated by all the networks, and the predictions generated by the correct network, and weighting those such that networks with closer spatial distances to the correct network should have closer predictions, whereas, networks that are further away from the correct network have more leeway and should have further estimates (Step 3 - Blue). We can create inductive biases across the ensemble via weight-sharing across the initial layers (Appendix A.3), which improves performance (Appendix A.6)

For the R-Block, we minimize sMAPE (Symmetric Mean Absolute Percentage Error) as the primary objective in a forecasting task:

$$\text{sMAPE} = \frac{1}{N} \sum_{i=1}^{N} 2 \times \frac{|\boldsymbol{y_i} - \hat{\boldsymbol{y}}_{\boldsymbol{i}}|}{|\boldsymbol{y_i}| + |\hat{\boldsymbol{y}}_{\boldsymbol{i}}|} \tag{13}$$

where N is the number of examples used for training. sMAPE is a metric that has been typically reported in the past with competitions like the M4 time series forecasting competition Makridakis et al. (2018). For the S-Block, we utilize a standard cross-entropy loss for multi-class classification.

The goal of the S-block is to translate a high-frequency time series into a spatial coordinate system with a dimensionality of a, b. The flexibility of fully connected networks in conjunction with spatial constraints imposed by convolutional filters biases the network towards a spatial representation of the temporal networks.

The S-block consists of four key layers, a 1D-CNN, a fully connected network layer with $(a + 2) \times (b + 2)$ number of units, a layer that reshapes the fully connected network block into an $(a + 2) \times (b + 2)$ rectangle, followed by a 3×3 convolution with a stride length of 2, to produce an ultimate output layer of dimension ab (Figure 5).

A discrete state space was chosen to improve the interpretability of the model sub-networks to produce meaningful results. However, future work may replace the discrete state space output with a representation of a density distribution or a continuous vector space.

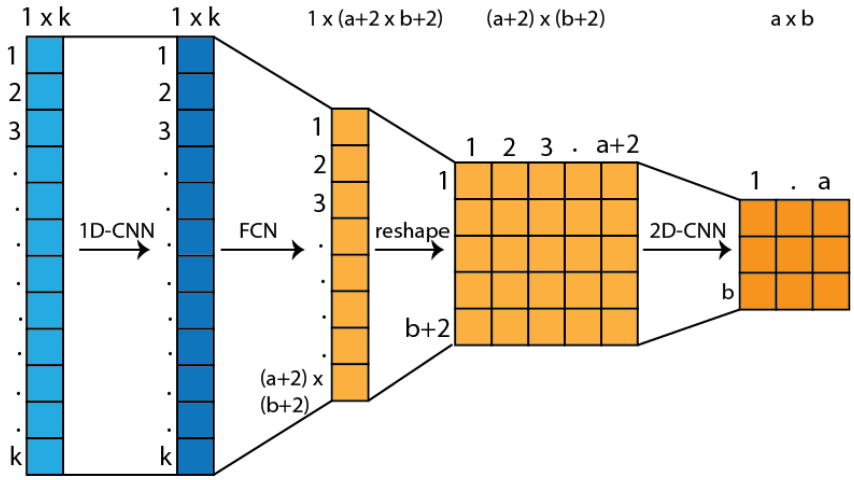


Figure 5: Network architecture of the S-Block. The S-block consists of four key layers, a 1D-CNN, a fully connected network layer with $(a + 2) \times (b + 2)$ number of units, a layer that reshapes the fully connected network block into an $(a + 2) \times (b + 2)$ rectangle, followed by a 3×3 convolution with a stride length of 2, to produce an ultimate output layer of dimension width length.

In order to compare the distinct properties of each state and demonstrate the ability to cluster waveforms within a given state, we train the S-Block with a latent state space of 3×3. We train a KNN with $k = 9$ on the 9 $(3 \times 3 = 9)$ distinct waveforms aggregated by state, and evaluate its ability to cluster all the waveforms on the dataset using a silhouette score, which is widely used to evaluate

the goodness of a clustering technique. SpaRTEn outperforms all other methods used to cluster time series on all four datasets.

A.8 Clinical Interpretation of Intracranial Pressure Waveforms

At point (0,0), the waveform is both stable and relatively constant. This indicates that the intracranial pressure does not require some form of intervention, and is a strong baseline for what non-pathological activity should look like. At point $(0, 1)$ we start seeing evidence of pathological neuro-vascular activity - the mean change in the intracranial pressure is relatively small, unlike the variation, which is relatively large. ICP variability is part of the response to injuries like trauma Svedung Wettervik et al. (2020). Following trauma, for example, high intracranial pressure variability is a physiological response, and suggests that some of the compensatory mechanisms are starting to be hyper-reactive. In contrast, at point $(2, 0)$, intracranial pressure waveform has a U-shaped, which indicates that the net change in the intracranial pressure over this time series is zero. However, the visibly slower dip and return to baseline suggests a relatively slow, hypo-reactive compensatory response to changes in intracranial pressure. A hypo-reactive intracranial pressure is associated with worse performance on the Glasgow Coma Scale Tian et al. (2013). A physician may try to shift a patient's state from $(0, 2)$ to $(0, 1)$ in order to improve outcomes by increasing the tone of the sympathetic nervous system Schmidt et al. (2018).

In the second row, we start to see acute cerebro-vascular dysregulation. At point $(1, 1)$, we see that the there are signs of instability, followed by a complete over-compensation, and an acute drop in the intracranial pressure. A patient in this state may warrant a CT scan to detect an early aneurysmal rupture. At point (1,1) and (1, 2), we see that there is complete dysregulation of the brain's vasculature with dramatic decreases and increases in intracranial pressure, respectively. These two waveforms are adjacent to each other and highlight intracranial pressure waveforms in a pathological state. In the context of waveform $(1, 2)$, we might clinically witness a hemorrhage. A hemorrhage can increase local volume of blood, and decrease intracranial pressure. If a patient is in a hemorrhagic state such as an intracranial hemorrhage, interventions include endotracheal intubation to protect the airway, blood pressure management and hypertonic saline to reduce intracranial pressure Caceres & Goldstein (2012). In $(1, 1)$, we see a rapid decrease in intracranial pressure as might be expected following treatment such as placement of an extra-ventricular drain Kramer (2021). Notably, the waveform in $(1, 1)$ is closer to the baseline state than that in $(1, 2)$, which makes sense because $(1, 1)$ involves a treatment designed to restore physiologic state.

The bottom row, namely $(2, 0)$ and $(2, 1)$ represents decompensation but more chronically than acutely as was observed with $(1, 1)$ and $(1, 2)$. In $(2, 0)$, we could see what chronic hypotension could look like, whereas in (2, 1), we might see what chronic hypertension would look like. Intracranial hypotension is associated with headaches Luetzen et al. (2021), and can either be acute or chronic. In $(2, 1)$ we notice that there is some form of chronic hypertension, which can be treated clinically with a diuretic drug. In $(2, 2)$, we see instability with respect to intracranial pressure, which can be a precursor to $(1, 2)$ and $(1, 1)$ Oernbo et al. (2022). These analyses demonstrate that SpaRTEn is able to decipher clinically meaningful states. Moreover, utilizing these state analyses to better disentangle states can improve the understanding of clinical treatment and associated outcomes Samartsidis et al. (2018).

Table 6: Silhouette Score of representations of Intracranial Pressure Waveforms

	Sample Size			
Model	10	25	100	400
Spectral	0.131 ± 0.023	0.165 ± 0.008	0.109 ± 0.007	0.156 ± 0.003
Random	0.366 ± 0.018	0.275 ± 0.007	0.252 ± 0.005	0.276 ± 0.002
GMM	0.341 ± 0.019	0.345 ± 0.005	0.329 ± 0.004	0.344 ± 0.003
SpaRTeN	$\mathbf{0.415} \pm 0.020$	$\mathbf{0.422} \pm 0.006$	$\mathbf{0.385} \pm 0.005$	$\mathbf{0.405} \pm 0.002$

To our knowledge, this is the first algorithm to use an online contrastive learning approach for time series classification. While it is known that discriminative region-based zero-shot learning in images can preserve context information Narayan et al. (2021), SpaRTeN representations quantitatively

capture variations within the data, and qualitatively provide key clinical insights into waveform patterns.

A.9 Potential Limitations

SpaRTeN is a novel min-max framework for decoding states, and has many of the same advantages and disadvantages as other min-max frameworks. Without sufficient gradient-based optimizations like smoothing and replacing density-based losses with negative log-likelihood losses, the gradients and states learned by SpaRTeN can be highly unstable (Appendix A.3). Subsequently, a collapse in the gradients on one of the blocks can be highly detrimental to other blocks.

Second, many datasets, especially in the ICU contain multi-modal sources of information. Currently, models like temporal fusion transforms can better account for multi-modal trends in time series and combine categorical with continuous variables. We anticipate further development of the SpaRTeN framework by including R-blocks that are capable of accounting for different variable types and data modalities may further enhance the ability of SpaRTeN to generate multi-modal archetype waveforms, which can be subsequently used to qualitatively evaluate changing states in the clinical setting.

Third, we selected 2D geometry because it was computationally tractable in terms of the distance-weighted contrastive loss, and interpretable in the ICU setting. Ablation of the the distance-weighted contrastive loss leads to poorer representation learning and clustering. Future work could explore higher-dimensional latent spaces and hyperbolic geometry.