

# COPER: Continuous Patient State Perceiver

Vinod Kumar Chauhan

*Institute of Biomedical Engineering,  
University of Oxford  
Oxford, UK*

Email: vinod.kumar@eng.ox.ac.uk

Anshul Thakur

*Institute of Biomedical Engineering,  
University of Oxford  
Oxford, UK*

Email: anshul.thakur@eng.ox.ac.uk

Odhran O'Donoghue

*Institute of Biomedical Engineering,  
University of Oxford  
Oxford, UK*

Email: odhran.odonoghue@eng.ox.ac.uk

David Andrew Clifton

*Institute of Biomedical Engineering,  
University of Oxford  
Oxford, UK*

Email: davidc@robots.ox.ac.uk

**Abstract**—In electronic health records (EHRs), irregular time-series (ITS) occur naturally due to patient health dynamics, reflected by irregular hospital visits, diseases/conditions and the necessity to measure different vitals signs at each visit etc. ITS present challenges in training machine learning algorithms which mostly are built on assumption of coherent fixed dimensional feature space. In this paper, we propose a novel COntinuous patient state PERceiver model, called COPER, to cope with ITS in EHRs. COPER uses Perceiver model and the concept of neural ordinary differential equations (ODEs) to learn the continuous time dynamics of patient state, i.e., continuity of input space and continuity of output space. The neural ODEs help COPER to generate regular time-series to feed to Perceiver model which has the capability to handle multi-modality large-scale inputs. To evaluate the performance of the proposed model, we use in-hospital mortality prediction task on MIMIC-III dataset and carefully design experiments to study irregularity. The results are compared with the baselines which prove the efficacy of the proposed model.

**Index Terms**—electronic health records, irregular time-series, neural ordinary differential equations, deep learning, Perceiver, continuous embedding.

## I. INTRODUCTION

Irregular time-series (ITS) are the results of uneven time intervals between different measurements, which lead to missing values [1] and can also create sparsity. In healthcare, ITS occur due to several reasons such as irregularity in the patient visits itself, dependence of physiological measurements at each visit on the patient health status and availability of staff to take measurements etc. [2]. ITS are prevalent in primary care as well as secondary care, including critical care, e.g., the MIMIC-III dataset has a general missing rate of 85% rising to over 90% for hourly sampling rate data [3].

This work was supported in part by the National Institute for Health Research (NIHR) Oxford Biomedical Research Centre (BRC) and in part by InnoHK Project Programme 3.2: Human Intelligence and AI Integration (HIAI) for the Prediction and Intervention of CVDs: Warning System at Hong Kong Centre for Cerebro-cardiovascular Health Engineering (COCHE).

Vinod Kumar Chauhan is supported by a Medical Research Council (MRC) Research Grant (MR/W01761X/1).

The views expressed are those of the authors and not necessarily those of the NHS, the NIHR, the Department of Health, the InnoHK – ITC or the MRC.

Use of electronic health records (EHRs) have greatly helped to develop machine learning (ML) models to guide clinical decision making, reduce the workload on an already burdened system and increase efficiency of healthcare resources [4], [5]. However, ML models are mostly based on the assumption of coherent fixed-dimensional feature space, which is invalidated by irregularity in the EHRs, and it is challenging to train ML models without affecting performance [6]. Thus, it is crucial for several reasons, including diagnosis, prognosis, treatment and resource management, to develop techniques for the accurate estimation of missing time-steps.

Due to the wide prevalence of ITS in EHRs and its importance, it have received increasing attention and there has been extensive research in recent years to deal with irregularity, e.g., [1], [2], [7], [8]. A wide variety of techniques have been developed to model ITS, e.g., simple statistical techniques for replacement, such as mean value, interpolation, imputation and matrix completion-based techniques [9] to modification of recurrent neural networks [10], neural ODEs [7], neural processes [2] and attention based techniques [1] etc.

The conventional simple statistical techniques for replacement, such as mean, median, zero and carry forward are biased and make strong assumptions about the data generating process, and results in loss of performance in downstream prediction tasks [11]. Many other modern techniques for modelling fail to capture feature-correlations in time-series [10], separate the modelling of missingness from the downstream task, and fail to learn the missingness pattern, or are not adequately efficient to handle large-scale inputs and multi-modality of the data [9] etc. Moreover, there are few techniques that can handle completely missing time steps, such as in [7], [12] but others that can only handle partially missing values, such as in [1].

In this paper, we propose a novel COntinuous patient state PERceiver model (COPER), for end-to-end learning from the ITS in EHRs, which can handle completely missing time steps. COPER uses neural ODEs to obtain continuous time dynamics of patient state from which completely missing steps can be calculated to feed to the Perceiver model [13]. The Perceiver

model learns the temporal dynamics of the time-series and has the capability to deal with large-scale multi-modality inputs.

The first novelty of our work is in learning continuous patient states using embedding and neural ODEs. This is different from the literature where ODEs are used for the obtaining the continuity of the latent representations [7], [12]. The continuous patient states could be helpful in a wide range of tasks including diagnosis, prognosis and treatment and in disease progression modelling. The second novelty of our work, to the best of our knowledge, is in adapting the Perceiver model [13] to time-series data. Perceiver models are recent advancements in transformer models [14] and are capable of working with large-scale multi-modality inputs. This could be very helpful in healthcare as EHRs represent big data about the patients, and taking into consideration the complete and long trajectory of patient states can yield good results for patient care. Similar to [7], we also use neural ODEs for the continuity of latent representations of the Perceiver model. Thus, COPER has a double engine – in terms of continuity in input and output – enabling it to cope well with the ITS in EHRs. We present the results of an in-hospital mortality task using the MIMIC-III dataset to demonstrate the utility of the proposed model.

## II. BACKGROUND

### A. Perceiver

Perceiver [13] based models are recent advancements to transformer [14] models which solve the issue of quadratic dependence of transformers on the input by introducing cross-attentions of inputs with a learnable tight latent vector. The cross-attention distills the high dimensional input to low a dimensional latent which is followed by self-attentions on the latent vector, as given below. Suppose,  $X_c \in \mathbb{R}^{n \times t' \times e}$  is a time-series data with  $n$  patients,  $t'$  time steps each represented with an embedding of size  $e$ , and  $Z \in \mathbb{R}^{n \times t'' \times d}$  is a latent vector (of size  $t'' \times d$  but repeated to match batch dimension) with  $t''$  number of latents ( $t'' \ll t'$ ) and  $d$  is latent dimension, then cross-attention and self-attention operations of the Perceiver can be represented as:

$$Z = f_{\text{CROSS-ATTEND}}(X_c, Z), \quad (1)$$

$$Z = f_{\text{SELF-ATTEND}}(Z), \quad (2)$$

where cross-attention is calculated using a scaled dot-product attention [14] using  $Z$  as query and  $X_c$  as key and values. Self-attention is also calculated similarly but with same input as query, key and value. In Perceiver [13], each cross-attention is followed by more than one self-attention layers which forms one block. There are several blocks of cross- and self-attention layers, each of which take the same input. The Perceiver based models are multi-modality models which can take large inputs and capture long contexts. There have been several other improvements to Perceiver [13], e.g., Perceiver IO [15] generalizes the outputs by introducing cross-attentions at the output similar to at input, and Perceiver AR [16] extends Perceivers to auto-regressive modelling by introducing masking for attentions etc.

### B. Neural Ordinary Differential Equations

Neural ordinary differential equations (ODEs) [12] are also a recently developed class of neural networks which help to capture the dynamics of a hidden state using ODEs, resulting into a continuous state space. Neural ODEs consist of a neural network which outputs derivative of the hidden state. The derivative of hidden state is fed to a black-box ODE solver which allows to calculate the hidden state at any time step, resulting in a continuous space, as given below.

$$\frac{dZ}{dt} = f_{\theta}(Z(t), t), \quad (3)$$

$$Z_0, \dots, Z_N = \text{ODESOLVER}(f_{\theta}, Z_0, (t_0, \dots, t_N)), \quad (4)$$

where  $Z$  is hidden state,  $f_{\theta}$  is a neural network which parameterizes the derivative of hidden state, i.e., continuous time dynamics. The ODESolver takes the derivative and initial hidden state  $Z_0$  and calculates the hidden state at  $(t_0, \dots, t_N)$ .

The concept of neural ODEs is extended to different domains for solving different problems, including solving the irregular time-series problem. [12] discussed a case to fit and extrapolate irregular time-series, which was then extended by [7] to handle ITS by modelling the hidden state dynamics of RNNs using neural ODEs. Similarly, there are other such extensions, such as those in [17] to handle ITS.

## III. METHODOLOGY

COPER has the best of neural ODEs and Perceiver networks to handle the ITS in EHRs, and can be applied to different tasks; here, we apply COPER to an in-hospital mortality prediction task. The overall working of COPER is represented in Fig. 1 using a univariate ITS. It consists of two neural ODEs, one each for the continuity of the input and the output, one Perceiver to learn the hidden time-series representations, and two multi-layer perceptrons (MLPs); one for learning embedding for patient state at each time step and one for the classification task, as discussed below.

### A. Embedding

Embeddings are representations of the input and can be used to adjust the size of the input. Here, the embedding layer is implemented with an MLP with single linear layer of 32 neurons which takes input as a time step (which is a vector of all features/variables at the given time) and produces embedding for the patient state at that point of time, as represented below:

$$X_e = f_{\text{EMB}}(X), \quad (5)$$

where  $X \in \mathbb{R}^{n \times t \times i}$  is input ITS data and  $X_e \in \mathbb{R}^{n \times t \times e}$  is a learned embedding for  $n$  patients,  $t$  time steps in the sequence,  $i$  input features/variables and  $e$  is the size of the embedding.

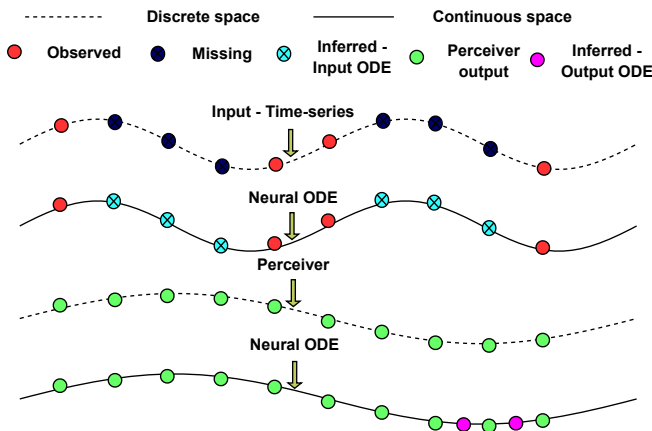


Fig. 1. Dynamics of COPER: Neural ODE generates a continuous space from the ITS with missing steps, from which missing observations can be calculated and fed to Perceiver. The discrete latent states generated by Perceiver are again passed to another neural ODE that makes the output space continuous allowing COPER to predict patient condition at any time point.

### B. Input Neural ODE

A neural ODE, immediately after the embedding layer, helps to make the input space continuous and learns the patient state dynamics from the embedding. This is expected to be less complex to learn compared with learning the hidden state dynamics of recurrent networks in latent ODE [7] because patient states, mostly, change slowly from one time step to another. This layer could be helpful to generate regular time-series, to work with different ML algorithms and can be independently plugged to other models, and further studied for disease progression modelling.

$$X_c = f_{\text{ODE-IN}}(X_e), \quad (6)$$

where  $X_c \in \mathbb{R}^{n \times t' \times e}$  is a regular time-series generated from embedding of ITS and  $t'$  is the sequence length.

### C. Perceiver

Perceivers are recent advancements to Transformers [14], as discussed in II-A. In our architecture, we have used one cross-attention followed by one self-attention, which can be represented as

$$Z = f_{\text{PER}}(X_c), \quad (7)$$

where  $Z \in \mathbb{R}^{n \times t'' \times d}$  is a latent vector with  $t''$  number of latents ( $t'' \ll t'$ ) and  $d$  is latent dimension. Perceiver with an input neural ODE can have lower number of latents but Perceiver needs latents equal to sequence length to have output ODE because it needs to mask the input. Perceiver outputs a set of discrete hidden representations, as shown in Fig. 1.

### D. Output Neural ODE and Classifier

Similar to input ODE, output ODE helps to learn the time continuous dynamics of the hidden state of Perceiver and is also helpful to deal with irregular time-series and can be used to predict future risk of patient deterioration. Thus, COPER

provides a double engine to handle ITS. It can be represented as:

$$Y_c = f_{\text{ODE-OUT}}(Z), \quad (8)$$

where  $Y_c \in \mathbb{R}^{n \times t'' \times d}$  represents continuous hidden state. Classifier layer is task specific and in this case it is a linear layer with one neuron to predict in-hospital mortality task.

### E. Difference from Existing Work

Our work has some similarity to [7] and [13]. However, our work employs neural ODEs for the continuity of input as well as output, unlike [7] which applies to only hidden state of RNNs. Moreover, [7] used encoder-decoder architecture with an ODE-RNN as encoder and ODE as a decoder. We have adapted the idea of Perceiver [13] to time-series and we use single cross-attention followed by one self-attentions, which require use of masking to avoid leaking future information. Moreover, unlike [13], we do not repeat cross- and self-attention blocks with the same input, which helps to reduce complexity, but still learns well.

## IV. RESULTS

### A. Experimental Settings

All the experiments are implemented in Pytorch and executed on an Ubuntu machine (64GB RAM, 1 NVIDIA GeForce GPU). Parameters of COPER are selected using trial and error as: 32 neurons in single embedding layer, 3 hidden layers of 128 neurons for each neural ODE, 128 cross- and self-attend head dimension, 64 latent dimension, dropout of 0.5 for attentions, neural ODE network as well as for LSTM. LSTM has two layers with hidden state size 50. We have used Adam optimizer with a constant learning rate of 0.0001. In addition to dropout, we use early stopping with a patience of 10 epochs to avoid overfitting. All experiments are executed three times with different seed values. The code to reproduce the results is publicly released at <https://github.com/jmdvinodjmd/COPER>.

### B. Dataset, Task and Baselines

We used the publicly available MIMIC-III dataset [18] for an in-hospital mortality prediction task, which contains time-series data in the ICU setting. Following [19], we use a dataset with 76 feature variables and 14,681, 3,236 and 3,222 samples in train, validation and test datasets. In-hospital mortality is a binary classification task to predict from the first 48 hours of ICU admission for hourly data if patient will die in the hospital or not. Mortality prediction is very important for triage, initial risk assessment, resource management and designing effective treatment plans [10]. We take LSTM and Perceiver as baselines to compare with COPER at 0%, 25%, 50% and 75% removal of time steps. Since in real-life, there could be large gaps between hospital visits so we have removed the data in chunks, and to keep experiments simple, we removed in three chunks, i.e., after first observation, from middle and from end. For baselines, we employ a carry-forward technique for the missing steps, while COPER uses neural ODEs to calculate the missing steps.

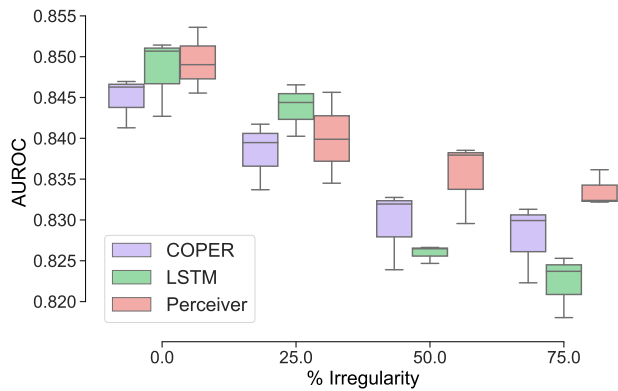


Fig. 2. Comparison of COPER with baselines – COPER copes comparatively with irregularity.

### C. Results

Fig. 2, compares COPER against the baselines, i.e., LSTM and Perceiver using AUROC by removing 0%, 25%, 50% and 75% time steps. From the results, we can make following inferences. First, comparing the models at 0% removal of time-steps, i.e., at normal training, we observed that Perceiver outperforms others and thus can provide another alternative to recurrent networks for processing time-series data. COPER performed lesser than the Perceiver, this could be because of the difficulty in the optimization (which will be explored in the future work) as two neural ODEs are attached to Perceiver in COPER. It is noted that LSTM outperforms other models at 25% removal of time-steps but as more time-steps are removed, it performs the worst. Interestingly, Perceiver with carry-forward handles the irregularity quite well at 50% and 75%. This could be because of the fact that it is a powerful model and carry-forward performs quite well in healthcare setting to deal with missingness [20]. Finally, we observed that COPER performs comparatively with Perceiver at 25% irregularity and has lesser percentage drop, also COPER outperformed LSTM at 50% and 75% irregularity.

Thus, the experimental evaluation shows that the Perceiver could be an alternative to recurrent networks, and COPER copes comparatively with ITS in healthcare.

### V. CONCLUSION AND FUTURE SCOPE

In this paper, we used embedding and neural ODE to learn continuous patient states, and adapted Perceiver model for time series to develop a model, called COPER, to cope with irregular time-series in healthcare. COPER has continuous input and output space which helps to solve the irregularity issue, and from the Perceiver, it inherits the capability to deal with multi-modality large-scale long context inputs.

The continuity of patient state is helpful to calculate the patient's status at times when patient measurements are not available. Similarly, continuous hidden state of Perceiver allows COPER to make predictions in continuous time irrespective of the input sampling frequency. The limitation of this work is that we have evaluated the proposed framework on only

MIMIC-III. In future, we will do extensive experimentation using more datasets for more rigorous performance evaluation.

### REFERENCES

- [1] Y. Lee, E. Jun, J. Choi, and H.-I. Suk, "Multi-view integrative attention-based deep representation learning for irregular clinical time-series data," *IEEE Journal of Biomedical and Health Informatics*, pp. 1–1, 2022.
- [2] P. Sharma, F. E. Shamout, V. Abrol, and D. A. Clifton, "Data pre-processing using neural processes for modeling personalized vital-sign time-series data," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 4, pp. 1528–1537, 2022.
- [3] C. Sun, S. Hong, M. Song, and H. Li, "A review of deep learning methods for irregularly sampled medical time series data," *arXiv preprint arXiv:2010.12493*, 2020.
- [4] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, "Deep ehr: a survey of recent advances in deep learning techniques for electronic health record (ehr) analysis," *IEEE journal of biomedical and health informatics*, vol. 22, no. 5, pp. 1589–1604, 2017.
- [5] A. Thakur, P. Sharma, and D. A. Clifton, "Dynamic neural graphs based federated reptile for semi-supervised multi-tasking in healthcare applications," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 4, pp. 1761–1772, 2021.
- [6] B. M. Marlin, D. C. Kale, R. G. Khemani, and R. C. Wetzel, "Unsupervised pattern discovery in electronic health care data using probabilistic clustering models," in *Proceedings of the 2nd ACM SIGHIT international health informatics symposium*, 2012, pp. 389–398.
- [7] Y. Rubanova, R. T. Chen, and D. K. Duvenaud, "Latent ordinary differential equations for irregularly-sampled time series," *Advances in neural information processing systems*, vol. 32, 2019.
- [8] S. N. Shukla and B. Marlin, "Multi-time attention networks for irregularly sampled time series," in *International Conference on Learning Representations*, 2021.
- [9] J. Yoon, W. R. Zame, and M. van der Schaar, "Estimating missing data in temporal data streams using multi-directional recurrent neural networks," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 5, pp. 1477–1490, 2018.
- [10] Q. Tan, M. Ye, B. Yang, S. Liu, A. J. Ma, T. C.-F. Yip, G. L.-H. Wong, and P. Yuen, "Data-gru: Dual-attention time-aware gated recurrent unit for irregular multivariate time series," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 930–937.
- [11] R. J. Little and D. B. Rubin, *Statistical analysis with missing data*. John Wiley & Sons, 2019, vol. 793.
- [12] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," *Advances in neural information processing systems*, vol. 31, 2018.
- [13] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira, "Perceiver: General perception with iterative attention," in *International Conference on Machine Learning*. PMLR, 2021, pp. 4651–4664.
- [14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [15] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer *et al.*, "Perceiver io: A general architecture for structured inputs & outputs," *arXiv preprint arXiv:2107.14795*, 2021.
- [16] C. Hawthorne, A. Jaegle, C. Cangea, S. Borgeaud, C. Nash, M. Malinowski, S. Dieleman, O. Vinyals, M. Botvinick, I. Simon *et al.*, "General-purpose, long-context autoregressive modeling with perceiver ar," *arXiv preprint arXiv:2202.07765*, 2022.
- [17] P. Kidger, J. Morrill, J. Foster, and T. Lyons, "Neural controlled differential equations for irregular time series," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6696–6707, 2020.
- [18] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.
- [19] H. Harutyunyan, H. Khachatrian, D. C. Kale, G. Ver Steeg, and A. Galstyan, "Multitask learning and benchmarking with clinical time series data," *Scientific data*, vol. 6, no. 1, pp. 1–18, 2019.
- [20] M. Scherpf, F. Gräßer, H. Malberg, and S. Zaunseder, "Predicting sepsis with a recurrent neural network using the mimic iii database," *Computers in biology and medicine*, vol. 113, p. 103395, 2019.