Stochastic Sparse Sampling: A Framework for Local Explainability in Variable-Length Medical Time Series

Xavier Mootoo[1,2,3], **Alan A. Díaz-Montiel**[3], **Milad Lankarany**[3,4], **Hina Tabassum**[1,4]
xmootoo@my.yorku.ca, adiazmon@tcd.ie, milad.lankarany@uhn.ca,
hinat@yorku.ca

[1]York University
[2]Vector Institute
[3]University Health Network
[4]University of Toronto

Abstract

While the majority of time series classification research has focused on modeling fixed-length sequences, variable-length time series classification (VTSC) remains underexplored, despite its relevance in healthcare and various other real-world applications. Existing *finite-context* methods, such as Transformer-based architectures, require noisy input processing when applied to VTSC, while *infinite-context* methods, including recurrent neural networks, struggle with information overload over longer sequences. Furthermore, current state-of-the-art (SOTA) methods lack explainability and generally fail to provide insights for local signal regions, reducing their reliability in high-risk scenarios. To address these issues, we introduce *Stochastic Sparse Sampling* (SSS), a novel framework for explainable VTSC. SSS manages variable-length sequences by sparsely sampling fixed windows to compute localized predictions, which are then aggregated to form a final prediction. We apply SSS on the task of seizure onset zone (SOZ) localization, a critical VTSC problem requiring identification of seizure-inducing brain regions from variable-length electrophysiological time series. We evaluate SSS on the Epilepsy iEEG Multicenter Dataset, a heteregeneous collection of intracranial electroencephalography (iEEG) recordings, and achieve performance comparable to current SOTA methods, while enabling localized visual analysis of model predictions.

1 Introduction

Variable-length time series are prevalent throughout many areas of healthcare, including heart rate monitoring, blood glucose measurements, and electrophysiological recordings. Yet, the majority of time series classification (TSC) literature focuses solely on methods that process fixed-length sequences [1, 2]. *Finite-context* methods—such as Transformers, temporal convolutional networks (TCNs), multi-layer perceptrons (MLPs), and linear models—operate on fixed-length input segments but cannot inherently process variable-length sequences, which necessitates padding or truncation that distorts the input [3–17]. *Infinite-context* methods, including recurrent neural networks (RNNs), gated recurrent units (GRUs), long short-term memory networks (LSTMs), and state space models (SSMs), handle variable-length sequences, but can experience degradation and overcompression of information over long sequences [18–22]. Alternative approaches such as ROCKET and Dynamic Time Warping (DTW) with k-Nearest Neighbors (k-NN) address variable-length sequences but come at the cost of model expressivity, computational cost, and sensitivity to noise [23, 24].

At the same time, modern time series methods greatly lack model explainability with respect to local regions signal regions, lowering both their feasibility for critical applications and potential for

NeurIPS 2024 Workshop on Time Series in the Age of Large Models.

clinicial adoption [25]. Post-hoc explainability of specific signal segments—as opposed to full-signal analysis—is useful for uncovering important characteristics such as motifs, anomalies, or frequency variations. This is especially important in contexts where the relationship between pathology and signal characteristics is poorly understood, and as a result, providing insights into this relationship would greatly aid both clinicians and scientists.

This need for local explainability is crucial in seizure onset zone (SOZ) localization—the task of identifying the brain region where seizures originate—as effective treatment requires analysis of variable-length signals and understanding local signal characteristics [26]. The World Health Organization (WHO) reports epilepsy affects over 50 million people globally, establishing it as one of the most common yet poorly understood neurological disorders [27, 28]. Additionally, one-third of patients do not respond to antiepileptic drugs, making surgery the last resort and accurate SOZ localization essential for effectively planning the operation. The process of SOZ identification involves a two-step procedure: initial implantation of electrodes in areas suspected to contain the SOZ, followed by recording and visual analysis of intracranial electroencephalography (iEEG) signals by medical experts. The task of SOZ localization reduces to classifying individual electrode recordings, representing different regions within the brain. Effective localization of the SOZ is challenging due to the absence of clinically validated biological markers and the variable-length nature of iEEG signals—consequently, surgical success rates range from 30% to 70% [29, 30].

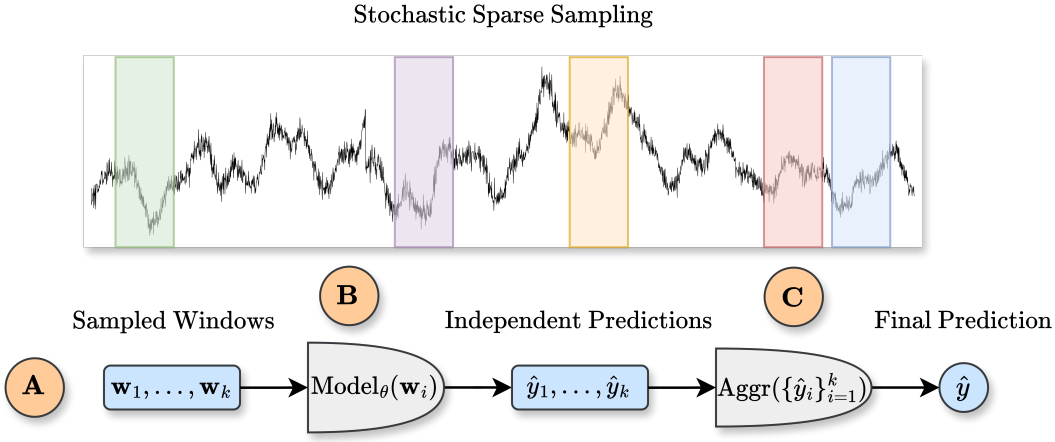

Figure 1: An overview of Stochastic Sparse Sampling (SSS). (**A**) For a given time series, we sample windows of fixed-length at random throughout the signal. (**B**) Each window is processed independently by a local model with parameters θ, outputting the individual predictions $\hat{y}_1, \ldots, \hat{y}_k$. (**C**) Individual predictions are then fed through an aggregation function to form the final prediction $\hat{y}$.

Contributions. We focus on the development of variable-length time series classification (VTSC) with local explainability, and propose a novel framework we call *Stochastic Sparse Sampling* (SSS). The main contributions of our paper are listed as follows:

- **Robustness to long and variable-length sequences**. SSS samples fixed-length windows, and processes them independently through a single model. This prevents context overload in long sequences seen in infinite-context methods, and does not utilize padding, truncation, or interpolation required by finite-context methods. By relying on a single local model, SSS utilizes far fewer parameters compared to finite-context methods that traditionally ingest the entire signal and are more susceptible to overfitting over long sequences.

- **Explainability through Local Predictions**: Our method enhances model interpretability by directly tying each output—a probability score for each window—to the overall prediction. This capability is crucial in critical clinical settings, such as SOZ localization, which traditionally relies on visual analysis. Given the significant risks associated with brain region removal, any proposal should be designed to integrate within clinical workflows. Moreover, in the absence of universally recognized biological markers for epilepsy, SSS offers the potential to further our understanding the SOZ and identifying novel markers.

- **Compatibility with modern backbones.** SSS seamlessly integrates with any time series backbone. This ensures that our approach leverages well-established frameworks, allowing for adaptability across a diverse array of contexts.

2 Method

2.1 Variable-length time series classification

Consider a collection of time series $\{(\mathbf{x}_t^{(1)})_{t=1}^{T_1}, \ldots, (\mathbf{x}_t^{(n)})_{t=1}^{T_n}\}$, where each series i has sequence length $T_i \in \mathbb{N}$, and for each time point t, the vector $\mathbf{x}_t^{(i)} \in \mathbb{R}^{M_i}$ has $M_i \in \mathbb{N}$ channels. The goal of VTSC is to learn an classifier f_θ which maps series $(\mathbf{x}_t^{(i)})_{t=1}^{T_i}$ to its corresponding class in $\{1, \ldots, K\}$ for $K \in \mathbb{N}$ classes. We require that the classifier f_θ can handle sequences of any length—that is, it has infinite context—since we assume that each T_i can be arbitrarily large at test time. Otherwise, we must adjust a finite-context classifier using padding, truncation, or interpolation. For a formal treatment of finite- and infinite-context methods, please see Appendix A.1.

2.2 Stochastic sparse sampling

Figure 1 provides an overview of SSS for a single time series. For each forward pass and each series $(\mathbf{x}_t^{(i)})_{t=1}^{T_i}$, SSS begins with a sampling procedure without replacement. Within the batch, a window from series i is selected with probability $p_i \approx T_i / \big(\sum_{j=1}^{n} T_j\big)$. This proportional sampling ensures fair representation of each series based on its length, allowing longer sequences—which contain more information—to contribute more samples. More formally, if N_i is the random variable representing the number of windows from series i in a batch of size B, then $N_i \sim \text{Binomial}(B, p_i)$, and consequently $\mathbb{E}[N_i] = Bp_i$. By sampling only a subset of windows, SSS introduces sparsity into the training process, reducing computational cost in memory and the likelihood of context overload.

After sampling a batch of windows $\{\mathbf{w}_1, \ldots, \mathbf{w}_B\}$, each $\mathbf{w}_b$ is processed independently by a model f_θ to obtain local prediction $\hat{y}_b = f_\theta(\mathbf{w}_b) \in [0,1]^K$ for $K \in \mathbb{N}$ classes. The choice of f_θ can be any time series backbone, in our experiments we select PatchTST [9]. Denote $I_b \in \{1, \ldots, n\}$ as the identifier of the time series from which $\mathbf{w}_b$ was sampled. To obtain the final prediction for time series $1 \leq i \leq n$, we aggregate the local predictions from all windows in the batch originating from series i:

$$\hat{y}^{(i)} = \frac{1}{N_i} \sum_{b=1}^{B} \hat{y}_b \cdot \mathbb{1}\{I_b = i\}, \tag{1}$$

where $N_i = \sum_{b=1}^{B} \mathbb{1}\{I_b = i\}$ is the number of windows in the batch for series i. By integrating local predictions directly into the final output, SSS allows for meaningful post-hoc explanations.

3 Experiments

3.1 Baselines

For our finite-context baselines, we include a variety of backbones including PatchTST [9], a patch-based Transformer model; TimesNet [14], which applies Fourier-based techniques along with inception TCN blocks [13]; ModernTCN [15], another TCN employing DWConv and FFNConv blocks; and DLinear [17], a linear neural network that utilizes seasonal-trend decomposition techniques. For our infinite-context baselines, we consider ROCKET [23], which applies random convolutional kernels with a linear head; DTW combined with k-NN [24]; and two RNN architectures: GRUs [19] and the SSM model Mamba [22]. For detailed configurations of each baseline see Appendix A.2.

3.2 Dataset

We utilize the Epilepsy iEEG Multicenter Dataset[1], consisting of iEEG signals with SOZ clinical annotations from four medical centers including the National Institute of Health (NIH), University of

[1] https://openneuro.org/datasets/ds003029/versions/1.0.7

Maryland Medical Center (UMMC), University of Miami Jackson Memorial Hospital (UMH), and Johns Hopkins Hospital (JHH). For each patient, and each electrode recording, the goal is to classify this univariate time series as SOZ or non-SOZ. This channel-independent formulation is primarily motivated due to interpatient variability, in our effort to build a general model that is unaffected by the varying number of channels per patient, and is thus more clinically applicable.

Table 1: SOZ localization. F1 score and AUROC are reported for each medical center, averaged over 5 seeds. For each center, we train and evaluate a separate model; the first column represents training and evaluation on all centers. **Red** and **blue** values indicate first and second best results respectively.

	All*		NIH		UMF		UMMC		JHH	
Model	F1	AUROC	F1	AUROC	F1	AUROC	F1	AUROC	F1	AUROC
SSS (Ours)	**0.7522**	**0.8009**	0.3984	0.6798	0.7227	0.9469	0.7616	0.8181	**0.7504**	**0.8331**
PatchTST [9]	0.7097	**0.7852**	**0.6509**	**0.7221**	**0.9386**	**0.9968**	**0.8085**	**0.8297**	0.7419	**0.8045**
TimesNet [14]	0.6897	0.7174	0.5950	0.6806	0.9227	0.9841	0.7821	0.8099	0.6891	0.8029
ModernTCN [15]	0.6938	0.7305	0.5055	**0.7220**	0.7221	**1.0000**	0.6371	0.8203	0.6710	0.7508
DLinear [17]	0.6916	0.7044	0.6055	0.6405	**0.9409**	0.9555	0.7658	0.7729	0.6873	0.7395
ROCKET [23]	0.6847	0.7481	**0.6520**	0.6546	0.8281	0.9327	0.7686	0.7900	0.6753	0.7752
DTW [24]	**0.7173**	0.7473	0.6210	0.7003	0.8080	0.8816	0.7370	0.7955	**0.7430**	0.7771
Mamba [22]	0.6452	0.7134	0.5974	0.6050	0.9289	0.9633	0.7900	**0.8424**	0.6456	0.6764
GRUs [19]	0.6948	0.7340	0.6171	0.6283	0.9044	0.9225	**0.7920**	0.8211	0.6140	0.6959

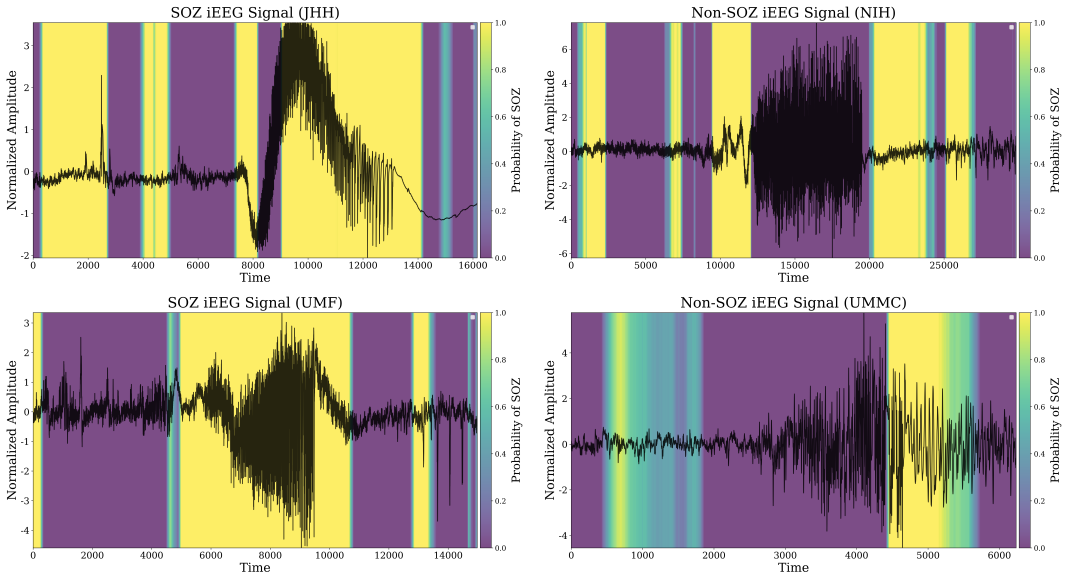

Figure 2: Visualization of SSS predictions using the PatchTST backbone with window size 1024, across patient iEEG channels within the SOZ (left) and non-SOZ (right). Normalized amplitude is displayed alongside a heatmap representing exponentially averaged window probabilities over time, where color intensity is proportional to the likelihood of the signal originating from the SOZ.

4 Discussion

Table 1 summarizes our experimental results for SOZ localization. Although SSS demonstrates subpar performance for certain clusters, it surpasses all SOTA baselines in the comprehensive all-cluster evaluation. This suggests SSS benefits from large and heterogeneous datasets, and effectively captures

local signal characteristics among electrophysiological recordings from different epilepsy patients. Figure 2 supports this claim, providing a visualization of SSS's predictions, where we observe clear qualitative differences in model predictions with respect to local signal characteristics, among all four medical centers. The ability to provide local explanation and visualization not only reinforces our method's potential for clinical application, but highlights its potential as a tool for fundamental science, to better understand the relationships between pathology and signal characteristics. For future work, we plan to improve cluster-specific performance and integrate uncertainty quantification into window predictions to enhance local explainability.

Acknowledgments and Disclosure of Funding

Xavier Mootoo is supported by Canada Graduate Scholarships—Master's (CGS-M) funded by the Natural Sciences and Engineering Research Council (NSERC) of Canada, the Vector Scholarship in Artificial Intelligence, provided through the Vector Institute, Canada, and the Ontario Graduate Scholarship (OGS) granted by the provincial government of Ontario, Canada. We also greatly thank Commune AI for generously providing the computational resources needed to carry out our experiments, in particular, we extend our gratitude to Luca Vivona and Sal Vivona. Hina Tabassum is supported through the NSERC Discovery Grant.

References

[1] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4): 917–963, 2019.

[2] Navid Mohammadi Foumani, Lynn Miller, Chang Wei Tan, Geoffrey I Webb, Germain Forestier, and Mahsa Salehi. Deep learning for time series classification and extrinsic regression: A current survey. *ACM Computing Surveys*, 56(9):1–45, 2024.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[4] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=rkgNKkHtvB`.

[5] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.

[6] 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.

[7] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proc. 39th International Conference on Machine Learning (ICML 2022)*, 2022.

[8] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International Conference on Learning Representations*, 2022.

[9] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

[10] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024.

[11] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 95–104, 2018.

[12] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[13] Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F Schmidt, Jonathan Weber, Geoffrey I Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962, 2020.

[14] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

[15] Donghao Luo and Xue Wang. Moderntcn: A modern pure convolution structure for general time series analysis. In *The Twelfth International Conference on Learning Representations*, 2024.

[16] Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O Arik, and Tomas Pfister. Tsmixer: An all-mlp architecture for time series forecasting. *arXiv preprint arXiv:2303.06053*, 2023.

[17] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? 2023.

[18] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1. 1986. *Biometrika*, 71(599-607):6, 1986.

[19] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[20] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[21] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

[22] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[23] Angus Dempster, François Petitjean, and Geoffrey I Webb. Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5): 1454–1495, 2020.

[24] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd international conference on knowledge discovery and data mining*, pages 359–370, 1994.

[25] Julia Amann, Alessandro Blasimme, Effy Vayena, Dietmar Frey, Vince I Madai, and Precise4Q Consortium. Explainability for artificial intelligence in healthcare: a multidisciplinary perspective. *BMC medical informatics and decision making*, 20:1–9, 2020.

[26] Sai Sanjay Balaji and Keshab K Parhi. Seizure onset zone identification from ieeg: A review. *IEEE Access*, 10:62535–62547, 2022.

[27] World Health Organization et al. *Epilepsy: a public health imperative*. World Health Organization, 2019.

[28] Carl E Stafstrom and Lionel Carmant. Seizures and epilepsy: an overview for neuroscientists. *Cold Spring Harbor perspectives in medicine*, 5(6):a022426, 2015.

[29] Wolfgang Löscher, Heidrun Potschka, Sanjay M Sisodiya, and Annamaria Vezzani. Drug resistance in epilepsy: clinical impact, potential mechanisms, and new innovative treatment options. *Pharmacological reviews*, 72(3):606–638, 2020.

[30] Adam Li, Chester Huynh, Zachary Fitzgerald, Iahn Cajigas, Damian Brusko, Jonathan Jagid, Angel O Claudio, Andres M Kanner, Jennifer Hopp, Stephanie Chen, et al. Neural fragility as an eeg marker of the seizure onset zone. *Nature neuroscience*, 24(10):1465–1474, 2021.

[31] Romain Tavenard, Johann Faouzi, Gilles Vandewiele, Felix Divo, Guillaume Androz, Chester Holtz, Marie Payne, Roman Yurchak, Marc Rußwurm, Kushal Kolar, et al. Tslearn, a machine learning toolkit for time series data. *Journal of machine learning research*, 21(118):1–6, 2020.

[32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[33] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2021.

[34] Daniele Grattarola, Lorenzo Livi, Cesare Alippi, Richard Wennberg, and Taufik A Valiante. Seizure localisation with attention-based graph neural networks. *Expert systems with applications*, 203:117330, 2022.

[35] Chunying Fang, Xingyu Li, Meng Na, Wenhao Jiang, Yuankun He, Aowei Wei, Jie Huang, and Ming Zhou. Epilepsy lesion localization method based on brain function network. *Frontiers in Human Neuroscience*, 18:1431153, 2024.

[36] Graham W Johnson, Leon Y Cai, Derek J Doss, Jasmine W Jiang, Aarushi S Negi, Saramati Narasimhan, Danika L Paulo, Hernán FJ González, Shawniqua Williams Roberson, Sarah K Bick, et al. Localizing seizure onset zones in surgical epilepsy with neurostimulation deep learning. *Journal of neurosurgery*, 138 (4):1002–1007, 2022.

[37] Bowen Yang, Baotian Zhao, Chao Li, Jiajie Mo, Zhihao Guo, Zilin Li, Yuan Yao, Xiuliang Fan, Du Cai, Lin Sang, et al. Localizing seizure onset zone by a cortico-cortical evoked potentials-based machine learning approach in focal epilepsy. *Clinical Neurophysiology*, 158:103–113, 2024.

A Appendix

A.1 Finite-context & infinite-context methods

Definition 1. Let $\mathcal{X}$ be a vector space over $\mathbb{R}$ and $f_\theta : \mathcal{X} \to \mathcal{Y}$ be a model with parameters θ and output space $\mathcal{Y}$. We say that f_θ has **finite-context** if $\mathcal{X}$ is finite-dimensional, that is, there exists some $n \in \mathbb{N}$ such that $\mathcal{X} \cong \mathbb{R}^n$ as vector spaces. Whereas f_θ is said to have **infinite-context** if $\mathcal{X} = \mathbb{R}^{(\infty)}$ is the space of real number sequences with finite support[2].

Note that this definition refers to the *native* capabilities of f_θ, without the usage of data manipulation techniques such as padding, truncation, and interpolation. We utilize this formalization to separate our baselines, so that we may better understand the advantages and limitations of both.

A.2 Experimental configurations and hyperparameter tuning

For each baseline in Table 1, we perform grid search and optimize with respect to best accuracy score on the all cluster evaluation. Most parameters are self-descriptive, and L refers to window or context size. The grid search parameters for each baseline are shown below; for information on the implementation of SSS, see Appendix A.4.

PatchTST: $d_{\text{model}} \in \{16, 32, 64\}$, $d_{\text{ff}} \in \{32, 64, 128\}$, `num_heads` $\in \{2, 4, 8\}$, `num_enc_layers` $\in \{1, 2, 3\}$, and `lr` $\in \{10^{-4}, 10^{-5}\}$, $L \in \{1000, 3000, 5000, 100000\}$. Best configuration: $d_{\text{model}} = 32$, $d_{\text{ff}} = 32$, `num_heads` $= 4$, and `lr` $= 10^{-5}$, and $L = 10000$. We adapt the official implementation `github.com/yuqinie98/PatchTST`, but swap out the attention module with the native PyTorch `torch.nn.MultiheadAttention` module.

TimesNet: $d_{\text{model}} \in \{16, 32, 64\}$, $d_{\text{ff}} \in \{32, 64, 128\}$, `num_kernels` $\in \{4, 6\}$, `top_k` $\in \{3, 5\}$, `num_enc_layers` $\in \{1, 2\}$, and `lr` $\in \{10^{-4}, 10^{-5}\}$, $L \in \{1000, 3000, 5000, 100000\}$. Best configuration: $d_{\text{model}} = 64$, $d_{\text{ff}} = 32$, `num_enc_layers` $= 2$, `num_kernels` $= 4$, `top_k` $= 3$, and $L = 5000$. We use the official implementation `github.com/thuml/Time-Series-Library`.

ModernTCN: `lr` $\in \{10^{-4}, 10^{-5}\}$, $d_{\text{model}} \in \{16, 32, 64\}$, `num_enc_layers` $\in \{1, 2\}$, `large_size_kernel` $\in \{9, 13, 21, 51\}$, `small_size_kernel` $= 5$, `dw_dims` $\in \{128, 256\}$, `ffn_ratio` $\in \{1, 4\}$, and $L \in \{1000, 3000, 5000, 100000\}$. Best configuration: `lr` $= 10^{-4}$, $d_{\text{model}} = 32$, `large_size_kernel` $= 9$, `dw_dims` $= 128$, `ffn_ratio` $= 4$, and $L = 10000$. We use the official repository `github.com/luodhhh/ModernTCN`.

DLinear: `moving_avg` $= 25$, `lr` $\in \{10^{-4}, 10^{-5}, 10^{-6}\}$, and $L \in \{1000, 3000, 5000, 100000\}$. Best configuration: `lr` $= 10^{1e-6}$ and $L = 10000$. We use the implementation from `github.com/thuml/Time-Series-Library`.

ROCKET: `num_kernels` $= 10000$ and `lr` $\in \{10^{-4}, 10^{-5}, 10^{-6}\}$. Best configuration: `lr` $= 10^{-4}$. We use the official implementation `github.com/angus924/rocket`.

DTW: `num_neighbors` $= 5$, `weighting` $=$ "uniform". We used the `KNNTimeSeriesClassifier` model from the `tslearn` library [31].

Mamba: `lr` $\in \{10^{-4}, 10^{-5}, 10^{-6}\}$, $d_{\text{model}} \in \{16, 32, 64\}$, `num_enc_layers` $\in \{1, 2, 3\}$. Best configuration: `lr` $= 10^{-6}$, `num_enc_layers` $= 2$, We also employ patching from [9], which we observed led to greater to performance, with `patch_size` $= 64$ and `patch_stride` $= 16$. We use the package `mambapy` which builds upon the official Mamba repository.

GRUs: Same search space used for Mamba, with added parameter of `bidirectional` $\in \{\text{True}, \text{False}\}$. Best configuration: `lr` $= 10^{-4}$, $d_{\text{model}} = 32$, `num_enc_layers` $= 3$, and `bidirectional` $=$ True. We utilized the native PyTorch module `torch.nn.GRU`.

In all experiments, we train using the Adam optimizer [32], for 50 epochs, with cosine learning rate annealing (one cycle with 50 epochs in length) which adjusts the learning rate down by two orders of magnitude (e.g., 10^{-4} to 10^{-6}) by the last epoch. We also implement early stopping with a patience of 15, and apply learnable instance normalization [33] for each input. For each experiment, we set the training, validation, test split is $70\%/10\%/20\%$. For most of the baselines we use a dropout rate of $0.2 - 0.3$, and weight decay to $10^{-4} - 10^{-5}$, but do not explicitly tune these parameters in our

[2]Every sequence in $\mathbb{R}^{(\infty)}$ must have finitely many non-zero terms.

grid search. For finite-context methods we set the batch size to the entire dataset (596 individual univariate time series for all clusters), whereas infinite-context methods required batch size of 1 due to their variable-length.

A.3 Data preprocessing

Due to the extremely long sequence length of many iEEG channels, we required downsampling to fit the dataset into memory (250GB RAM). For each channel, we applied a 1D average pooling layer with `kernel_size`=24 and `kernel_stride`=12 before feeding it to the baseline model or before performing the window sampling procedure for SSS. Due to the high computational complexity of DTW, we were required to further downsample to $\texttt{kernel\_size} = 60$ and $\texttt{kernel\_stride} = 30$.

A.4 SSS implementation and configurations

For SSS, our grid search space consisted of $\texttt{lr} \in \{10^{-4}, 10^{-5}\}$, $\texttt{batch\_size} \in \{2048, 4096, 8192\}$, $L \in \{512, 1024, 2048\}$, in addition to tuning the PatchTST backbone parameters as outlined in A.2. Best configuration: $\texttt{lr} = 10^{-4}$, $\texttt{batch\_size} = 8192$, $L = 1024$, $d_{\text{model}} = 32$, $d_{\text{ff}} = 128$, $\texttt{num\_heads} = 4$, and $\texttt{num\_enc\_layers} = 2$.

To achieve the sampling procedure described in 2.2, this is readily achieved by performing the slicing window method over all channels, and collecting all windows within a single dataset. Note that the number of windows obtained through the slicing window method will be proportional to the sequence length of the time series. During training, the native PyTorch dataloader samples windows uniformly, with sample size equal to our batch size, thus we achieve our chosen sampling procedure due to the method in which we constructed our dataset, resulting in $N_i \sim \text{Binomial}(B, p_i)$. This procedure uses sampling *without* replacement; one may consider replacement, however, we did not experiment with this and leave modifications with more complex sampling procedures as a future work.

A.5 SOZ localization specific methods

Several recent proposals have been tailored specifically to SOZ localization. Functional connectivity graphs compute patient-specific channel metrics to capture brain connectivity patterns [34, 35], offering insights into functional relationships associated with seizures. However, their reliance on intra-patient dynamics makes them unsuitable for a single model that can generalize across multi-patient, heterogeneous datasets. Alternatively, electrical stimulation methods that use intracranial electrodes [36, 37] can enhance localization accuracy through induced responses analyzed by TCNs and logistic regression models. Yet, these approaches require both fixed-length windows and the use of active stimulation. For our purpose of building a general model for SOZ localization, which can be applied to multiple patients (with a potentially varying number of channels) without electrical stimulation, we do not consider such approaches in our study.