RAxSS: Retrieval-Augmented Sparse Sampling for Explainable Variable-Length Medical Time Series Classification

Anonymous Author(s)
Affiliation
Address
email

Abstract

Medical time series analysis is challenging due to data sparsity, noise, and highly variable recording lengths. Prior work has shown that stochastic sparse sampling effectively handles variable-length signals, while retrieval-augmented approaches improve explainability and robustness to noise and weak temporal correlations. In this study, we generalize the stochastic sparse sampling framework for retrieval-informed classification. Specifically, we weight window predictions by within-channel similarity and aggregate them in probability space, yielding convex series-level scores and an explicit evidence trail for explainability. Our method achieves competitive iEEG classification performance and provides practitioners with greater transparency and explainability. We evaluate our method in iEEG recordings collected in four medical centers, demonstrating its potential for reliable and explainable clinical variable-length time series classification.

1 Introduction

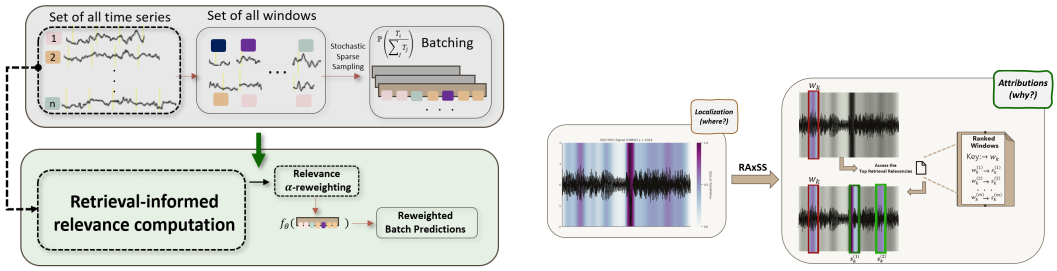
Artificial intelligence (AI) is increasingly embedded across clinical and translational workflows, with reported benefits for diagnostics, treatment planning, monitoring, and population health [1–3]. Nevertheless, routine deployment remains uneven. Two persistent barriers are the heterogeneity of clinical data and the need for transparent, clinician-oriented explanations [4].

One domain where the challenges are most keenly felt is medical-time series classification. Heart rate, glucose, and electrophysiology are examples of physiological signals that are both sparse and prone to noise, with their durations differing significantly between persons and events [5, 6]. Most of the time series classification (TSC) research, however, remains centered on approaches that operate with fixed-length sequences only [7–9].

Recently, Mootoo et al. [9] proposed the Stochastic Sparse Sampling (SSS) to address variable-length time series classification (VTSC). SSS samples fixed-length windows from long recordings, computes local predictions using a backbone model, and aggregates these to obtain a series-level decision [9]. While effective and computationally tractable, SSS aggregates window predictions uniformly, thereby treating all sampled segments as equally informative; its explainability relies primarily on visualizations of local scores. This assumption might especially be problematic in real-world time series, however, where non-stationary, irregular patterns may occur infrequently and lack strong temporal correlations, making generalization difficult [10, 11].

Retrieval-augmented methods address this by selectively leveraging similar past instances rather than memorizing all patterns. In time series, retrieval has also been explored across entities, where similarities guide aggregation for forecasting [12, 13]. Most recently, Retrieval-Augmented Forecasting of Time series (RAFT) [14] introduces a similarity-based retrieval mechanism for forecasting: it

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.



(a) **RAxSS pipeline.** Green box: conceptual addition to SSS (gray). Dashed boxes: retrieval steps.

(b) **Window ranking & attribution.** Ranked, non-identical neighbors explain each window's influence.

Figure 1: RAxSS workflow: (a) end-to-end pipeline and (b) retrieval-weighted explainable module.

retrieves past patches most similar to the current input and leverages their future continuations to improve predictions, with notable gains for rare patterns and weak temporal correlations

[14]. However, RAFT is tailored to forecasting and does not directly address TSC. Motivated by these advances, we propose RAxSS: Retrieval-Augmented Sparse Sampling for Explainable Physiological Time Series Classification, a variable-length time series classification (VTSC) framework that integrates a retrieval-informed relevance computation into the SSS pipeline. Using a medical usecase, we tackle the Seizure Onset Zone (SOZ) localization problem. More details about the problem can be found in the Appendix A.2.

RAxSS retains SSS's stochastic, length-proportional sampling and replaces uniform averaging with a similarity-weighted convex mix of window predictions. Using pearson or cosine similarity (as in Han et al. [14]), each window's top-m within-series neighbors define a support score that is softmax-normalized into aggregation weights. This design amplifies informative segments and downweights noisy ones. It also enables drill-down explanations: the series score is an additive sum of window contributions, each justified by a within-channel retrieval leaderboard.

Our contributions. (i) We provide a **methodological advance** for variable-length time-series classification by introducing a retrieval-weighted aggregation mechanism that ranks and weights windows within each series, thereby improving uniform averaging while preserving the efficiency of stochastic sampling. (ii) We align **explanation with aggregation** through a weighting scheme that produces quantitative, window-level attributions. These extend beyond static heatmaps and enable principled drill-down from series-level predictions to segment-level contributions. (iii) We demonstrate robustness in challenging regimes (the settings where retrieval excels for time-series modeling [14]), while retaining compatibility with diverse backbones, including transformer variants. RAxSS adapts retrieval mechanisms developed for forecasting the task of variable-length classification, combining stochastic coverage with similarity-guided prioritization to establish a framework that is selective, explainable, and well-suited to the demands of clinical time-series analysis.

2 Method

2.1 Datasets

Epilepsy iEEG Multicenter Dataset We use the Epilepsy iEEG Multicenter Dataset, comprising of intracranial EEG (iEEG) recordings with seizure onset zone (SOZ) from four centers: Johns Hopkins Hospital (JHH), the National Institutes of Health (NIH), University of Maryland Medical Center (UMMC), and University of Miami Jackson Memorial Hospital (UMH). Following the evaluation practice of [9], we report F1 score, area under the curve (AUC), and accuracy as primary metrics. Summary statistics and additional dataset and data-preprocessing details are provided in Appendix B.1.

2.2 Framework overview

RAxSS builds on SSS to handle variable-length medical time series by unifying sampling, retrieval, and aggregation in a single loop. As outlined in Fig. 1a and implemented in Alg. 1, long, noisy

recordings are segmented into fixed-length windows, sampled *length-proportionally* so that the probability of drawing from series i is $p_i \propto T_i / \sum_j T_j$, and scored by a backbone f_θ. In parallel (see Fig. 1b), a *within-series* retrieval computes Pearson or cosine similarities, forms for each query window the top-m *nonidentical* neighbors (which may temporally overlap due to sliding extraction), and summarizes their support. The retrieval-aware aggregator (Alg. 2) then converts these supports into softmax weights across windows and produces a *convex* series-level prediction by re-weighting and aggregating window-level outputs. Fig. 1b further illustrates the explanatory consequence: each influential window is accompanied by a ranked, possibly overlapping but nonidentical set of neighbors whose weights quantify *why* it matters. The result maintains SSS efficiency while incorporating resilient, retrieval-guided weighting and a clear evidence path from windows to the final prediction.

2.3 Enhancing explainability with RAxSS

A consequence of applying RAxSS is explainability, where we go beyond just localization but can also access the attributions. In more detail, for series i with window index set K_i, the base model outputs window posteriors $p_k = \mathrm{softmax}(z_k) \in \Delta^{C-1}$, $k \in K_i$.

For each $k \in K_i$, retrieve the m most similar nonidentical windows from the same series under $\phi \in \{\text{Pearson}, \text{Cosine}\}$:

$$s_k^{(j)} = \phi\big(w_k, w_j\big), \quad j \in N_k, \quad |N_k| = m, \bar{s}_k \quad = \frac{1}{m} \sum_{j \in N_k} s_k^{(j)} \tag{1}$$

Subsequently, we define window influence weights via a temperatured softmax over $\{\bar{s}_t\}_{t \in K_i}$:

$$\alpha_k \;=\; \frac{\exp(\bar{s}_k/\tau)}{\sum_{t \in K_i} \exp(\bar{s}_t/\tau)} \in [0,1], \qquad \sum_{k \in K_i} \alpha_k = 1\,. \tag{2}$$

Aggregation in probability space. Series-level probabilities are a convex combination of window posteriors (proof in Appendix A.1):

$$\hat{p}^{(i)} = \sum_{k \in K_i} \alpha_k \, p_k\,. \tag{3}$$

From "where?" to "why?" Explainability should go beyond localization and provide reasons for *why* specific regions are trusted. In Fig. 1b, the left panel shows the window-probability heatmaps of Mootoo et al. [9], which indicate *where* the model is confident. RAxSS adds attributions to answer the *why*: for each influential window k, we expose the evidence used to compute its weight. α_k by reporting (i) its summary support $\bar{s}_k$, the mean similarity to its top-m within-series neighbors, and (ii) a ranked neighbor leaderboard $\{(w_k^{(j)}, s_k^{(j)}) : j \in N_k\}$ with timestamps. These quantities explain why window k received a high contribution $\lambda_k \, p_{k,c}$ to the final series-level probability. Since

$$\frac{\partial \alpha_k}{\partial s_k^{(j)}} \;=\; \frac{1}{m\tau}\,\alpha_k\big(1-\alpha_k\big) \;>\; 0, \tag{4}$$

increasing any neighbor similarity strictly increases α_k (holding all other $\bar{s}_t$ fixed), making the leaderboard a faithful explanation of why w_k was weighted highly. For a selected channel we overlay: (a) the raw signal, (b) the window probability heatmap (*localization*), and (c) for the top-m supporting windows and their support $\alpha_{k,j}$ values (*attribution*). See Fig. 1b for an example visualization of our proposed explainability framework.

3 Experiments & Results

Results On multicenter iEEG, RAxSS is competitive with strong baselines (Table 1). The cosine variant achieves the best AUC (0.8046 ± 0.0346), edging the reproduced SSS (0.8035 ± 0.0686) and outperforming non-SSS baselines (e.g., PatchTST 0.7852). The Pearson variant yields higher F1 than cosine (0.7275 ± 0.0489 vs. 0.6967 ± 0.0791) and strong accuracy (70.51 ± 3.59), close to SSS (71.14 ± 6.31). Overall, cosine favors AUC, while Pearson offers a better F1/accuracy trade-off, letting practitioners pick the similarity to prioritize discrimination or balanced detection, while retaining built-in explainability. The training details are provided in Appendix 3.

Table 1: SOZ localization on **All** centers. F1, AUC, and Accuracy are averaged over **5 seeds**. For our runs (*RAxSS* variants and *SSS (reproduction)*), we used the *same seed set* and backbone code; the line *SSS (paper)* is the value reported by the original authors. Boldface values with * and † denote the best and second-best results per column, respectively.

Model	F1	AUC	Acc.(%)
RAxSS (cosine)	0.6967 ± 0.0791	$\mathbf{0.8046}^* \pm 0.0346$	69.76 ± 5.25
RAxSS (pearson)	$\mathbf{0.7275}^\dagger \pm 0.0489$	0.7980 ± 0.0537	$\mathbf{70.51}^\dagger \pm 3.59$
SSS (reproduction)	$\mathbf{0.7437}^* \pm 0.0537$	$\mathbf{0.8035}^\dagger \pm 0.0686$	$\mathbf{71.14}^* \pm 6.31$
SSS (Mootoo et al. [9])	0.7629	0.7999	72.35
PatchTST (Nie et al. [15])	0.7097	0.7852	66.83
TimesNet (Wu et al. [16])	0.6897	0.7174	65.98
ModernTCN (Luo and Wang [17])	0.6938	0.7305	68.42
DLinear (Zeng et al. [18])	0.6916	0.7044	68.41
ROCKET (Dempster et al. [19])	0.6847	0.7481	69.27
Mamba (Gu and Dao [20])	0.6452	0.7134	64.39
GRUs (Bahdanau et al. [21])	0.6948	0.7340	65.85
LSTM ([22])	0.6709	0.7144	65.43

4 Discussion & Conclusion

In this paper, our primary goal was to provide a more clinician-oriented, steerable and explainable framework for VTSC. To achieve this, we: (i) coupled stochastic sparse sampling with within-recording retrieval and probability-space aggregation; (ii) made explanations by exposing additive window contributions and an evidence leaderboard for influential windows; and (iii) preserved practicality via a model-agnostic, privacy-friendly design with simple knobs for steering. Results showed robust, competitive performance across centers, all while maintaining more transparency and explainability. RAxSS consistently ranks among the top approaches across metrics and sites, and we expect routine calibration and hyperparameter tuning to further boost absolute performance.

The window-based design already gives granular localizations by overlaying window-level probabilities (the *where*). To explain the *why*, we present, for each influential window, a ranked list of its (top-m) within-recording neighbors (nonidentical, overlap allowed) with their similarity scores and resulting weights. This is justified because a) similarities determine the window's weight via mean support and b) the cross-window softmax is strictly increasing in that support. Thus the same evidence that raises a window's weight justifies its contribution, yielding a faithfulness-oriented "why", on top of "where". Despite this transparency, finer-grained, mechanistic explanations will require probing internal representations and decision pathways.

In clinical use, inference is typically per recording, so length-proportional sampling offers no test-time benefit. Retrieval remains pivotal: it reweights window predictions by agreement with within-recording neighbors, improving robustness to noisy and idiosyncratic windows and providing inspectable evidence via the neighbor leaderboard. Over this medical setting, we couple the retrieval concept into time series classification (prior work emphasized forecasting [14]) enabling domain-aligned control to match clinical priorities.

Future work. Our current implementation performs retrieval and aggregation strictly within the same channel/recording. This choice (i) avoids dependence on cross-subject/center labels, (ii) reduces privacy exposure by not querying external data, and (iii) keeps the approach generic for other clinical time-series tasks. A natural extension is *pattern-level* retrieval: indexing canonical events (e.g., seizure onsets) and retrieving neighbors from the same subject or a curated, cross-center repository. While this may strengthen the quality of evidence and enable case-based reasoning, it requires additional curation/metadata and stronger governance (privacy and access control). Beyond scope, two technical directions are promising: learning the similarity/temperature parameters from data, and conducting comprehensive faithfulness stress tests (e.g., deletion/insertion tests, retrieval randomization, and counterfactual probes) to further validate the explanations.

References

[1] M. Sawkat Anwer. Opportunities & Challenges of Artificial Intelligent-Powered Technology in Healthcare. *Medical Research Archives*, 12(3), March 2024. ISSN 2375-1924. doi: 10.18103/mra.v12i3.5141. URL https://esmed.org/MRA/mra/article/view/5141.

[2] Ben Li, Dylan Powell, and Regent Lee. Commercialization of medical artificial intelligence technologies: challenges and opportunities. *NPJ digital medicine*, 8(1):454, July 2025. ISSN 2398-6352. doi: 10.1038/s41746-025-01867-w.

[3] Pranav Rajpurkar, Emma Chen, Oishi Banerjee, and Eric J. Topol. AI in health and medicine. *Nature Medicine*, 28(1):31–38, January 2022. ISSN 1546-170X. doi: 10.1038/s41591-021-01614-0. URL https://www.nature.com/articles/s41591-021-01614-0. Publisher: Nature Publishing Group.

[4] Qiyang Sun, Alican Akman, and Björn W. Schuller. Explainable Artificial Intelligence for Medical Applications: A Review. *ACM Trans. Comput. Healthcare*, 6(2):17:1–17:31, February 2025. doi: 10.1145/3709367. URL https://dl.acm.org/doi/10.1145/3709367.

[5] Dominik Walther, Johannes Viehweg, Jens Haueisen, and Patrick Mäder. A systematic comparison of deep learning methods for EEG time series analysis. *Frontiers in Neuroinformatics*, 17, February 2023. ISSN 1662-5196. doi: 10.3389/fninf.2023.1067095. URL https://www.frontiersin.org/journals/neuroinformatics/articles/10.3389/fninf.2023.1067095/full. Publisher: Frontiers.

[6] Elena Agliari, Adriano Barra, Orazio Antonio Barra, Alberto Fachechi, Lorenzo Franceschi Vento, and Luciano Moretti. Detecting cardiac pathologies via machine learning on heart-rate variability time series and related markers. *Scientific Reports*, 10(1):8845, June 2020. ISSN 2045-2322. doi: 10.1038/s41598-020-64083-4. URL https://www.nature.com/articles/s41598-020-64083-4. Publisher: Nature Publishing Group.

[7] 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, July 2019. ISSN 1384-5810, 1573-756X. doi: 10.1007/s10618-019-00619-1. URL http://arxiv.org/abs/1809.04356. arXiv:1809.04356 [cs].

[8] 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, December 2023. URL http://arxiv.org/abs/2302.02515. arXiv:2302.02515 [cs].

[9] Xavier Mootoo, Alan A. Díaz-Montiel, Milad Lankarany, and Hina Tabassum. Stochastic Sparse Sampling: A Framework for Variable-Length Medical Time Series Classification, October 2024. URL http://arxiv.org/abs/2410.06412. arXiv:2410.06412 [cs].

[10] 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. October 2021. URL https://openreview.net/forum?id=cGDAkQo1C0p.

[11] A. S. Weigend, M. Mangeas, and A. N. Srivastava. Nonlinear gated experts for time series: discovering regimes and avoiding overfitting. *International Journal of Neural Systems*, 6(4):373–399, December 1995. ISSN 0129-0657. doi: 10.1142/s0129065795000251.

[12] Tomoharu Iwata and Atsutoshi Kumagai. Few-shot Learning for Time-series Forecasting, September 2020. URL http://arxiv.org/abs/2009.14379. arXiv:2009.14379 [stat].

[13] Sitan Yang, Carson Eisenach, and Dhruv Madeka. MQRetNN: Multi-Horizon Time Series Forecasting with Retrieval Augmentation, September 2022. URL http://arxiv.org/abs/2207.10517. arXiv:2207.10517 [cs].

[14] Sungwon Han, Seungeon Lee, Meeyoung Cha, Sercan O. Arik, and Jinsung Yoon. Retrieval Augmented Time Series Forecasting, May 2025. URL http://arxiv.org/abs/2505.04163. arXiv:2505.04163 [cs].

[15] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers, March 2023. URL http://arxiv.org/abs/2211.14730. arXiv:2211.14730 [cs].

[16] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis, April 2023. URL http://arxiv.org/abs/2210.02186. arXiv:2210.02186 [cs].

[17] DongHao Luo and Xue Wang. ModernTCN: A Modern Pure Convolution Structure for General Time Series Analysis. *International Conference on Representation Learning*, 2024:31728–31770, May 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/86b1437c1e4c3b3c4debff98234a67e7-Abstract-Conference.html.

[18] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, volume 37 of *AAAI'23/IAAI'23/EAAI'23*, pages 11121–11128. AAAI Press, February 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i9.26317. URL https://doi.org/10.1609/aaai.v37i9.26317.

[19] Angus Dempster, François Petitjean, and Geoffrey I. Webb. ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels, October 2019. URL http://arxiv.org/abs/1910.13051. arXiv:1910.13051 [cs].

[20] Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces, May 2024. URL http://arxiv.org/abs/2312.00752. arXiv:2312.00752 [cs].

[21] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate, May 2016. URL http://arxiv.org/abs/1409.0473. arXiv:1409.0473 [cs].

[22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997. ISSN 1530-888X. doi: 10.1162/neco.1997.9.8.1735. Place: US Publisher: MIT Press.

[23] Sai Sanjay Balaji and Keshab K. Parhi. Seizure Onset Zone Identification From iEEG: A Review. *IEEE Access*, 10:62535–62547, 2022. ISSN 2169-3536. doi: 10.1109/ACCESS.2022.3182716. URL https://ieeexplore.ieee.org/document/9795000.

[24] Epilepsy: a public health imperative. URL https://www.who.int/publications/i/item/epilepsy-a-public-health-imperative.

[25] Carl E. Stafstrom and Lionel Carmant. Seizures and Epilepsy: An Overview for Neuroscientists. *Cold Spring Harbor Perspectives in Medicine*, 5(6):a022426, June 2015. ISSN 2157-1422. doi: 10.1101/cshperspect.a022426. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4448698/.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library, December 2019. URL http://arxiv.org/abs/1912.01703. arXiv:1912.01703 [cs].

A Technical Appendices and Supplementary Material

A.1 Proposition (convexity of the series-level probabilities).

Let K_i be the set of windows for series i. Assume each window posterior $p_k \in \Delta^{C-1}$ (entries nonnegative and summing to 1). Define

$$\alpha_k = \frac{\exp(\bar{s}_k/\tau)}{\sum_{t\in K_i}\exp(\bar{s}_t/\tau)}, \qquad \tau > 0.$$

Then $\hat{p}^{(i)} = \sum_{k\in K_i} \alpha_k\, p_k \in \Delta^{C-1}$, i.e., it is a *convex combination* of $\{p_k\}$.

Proof. Since $\exp(\cdot) > 0$, we have $\alpha_k \geq 0$ for all k, and by construction $\sum_{k\in K_i} \alpha_k = 1$. For each class c,

$$\hat{p}_c^{(i)} = \sum_k \alpha_k\, p_{k,c} \;\geq\; 0 \quad \text{because} \quad \alpha_k, p_{k,c} \geq 0.$$

Moreover,

$$\sum_{c=1}^{C} \hat{p}_c^{(i)} = \sum_c \sum_k \alpha_k\, p_{k,c} = \sum_k \alpha_k \big(\sum_c p_{k,c}\big) = \sum_k \alpha_k \cdot 1 = 1.$$

Thus $\hat{p}^{(i)}$ has nonnegative entries summing to 1, so $\hat{p}^{(i)} \in \Delta^{C-1}$ and, by definition, is a convex combination of the $\{p_k\}$. □

A.2 Seizure Onset Zone (SOZ) Localization problem description

Developing explainable methods for variable-length time series classification (VTSC) is especially critical in seizure onset zone (SOZ) localization, where clinicians must determine the brain regions that initiate seizures [23]. Epilepsy affects over 50 million people worldwide, making it one of the most prevalent but still poorly characterized neurological conditions [24, 25, 9]. For nearly one-third of patients, medication is ineffective, leaving surgery as the only option and placing high demands on accurate SOZ mapping. Current practice involves surgically implanting electrodes in candidate regions and visually inspecting intracranial EEG (iEEG) recordings to classify which channels correspond to the SOZ.

A.3 Algorithms

Algorithm 1: Variable Length Time Series Training Algorithm with Retrieval-augmented Aggregation (Single Epoch)

Input : Time series $\mathcal{X} = \{(x_t^{(1)})_{t=1}^{T_1}, \ldots, (x_t^{(n)})_{t=1}^{T_n}\}$;
Labels $\mathcal{Y} = \{y^{(1)}, \ldots, y^{(n)}\}$; model f_θ; batch size B; loss $\mathcal{L}$; .
Output : Updated parameters θ

$\mathcal{W} \leftarrow$ set of *all* windows from each series in $\mathcal{X}$
while $\mathcal{W} \neq \emptyset$ **do**
  `// Sample a minibatch of windows with length-proportional probabilities`
  $\mathcal{W}_0 \leftarrow \textsc{Sample}(\mathcal{W}, B)$ with $\Pr(\text{series } i) = \dfrac{T_i}{\sum_j T_j}$
  for $i = 1, \ldots, n$ **do**
    $\mathcal{W}_i \leftarrow \{\, w \in \mathcal{W}_0 \mid w \text{ comes from series } i \,\}$
    if $\mathcal{W}_i = \emptyset$ **then**
      continue
    `// Per-window retrieval signals for series` i
    foreach $w_k \in \mathcal{W}_i$ **do**
      $R_i[k] \leftarrow \textsc{Retrieve}(w_k, T_i)$
        `// a dictionary` $\{k_i^{(1)}{:}\rho_i^{(1)}, \ldots, k_i^{(m)}{:}\rho_i^{(m)}\}$ `of Pearson | Cosine scores`
    `// Window-level predictions`
    $\mathcal{Y}_i \leftarrow \{\, f_\theta(w) \mid w \in \mathcal{W}_i \,\}$
    `// Aggregate window predictions using retrieval signals`
    $\hat{y}^{(i)} \leftarrow \textsc{Aggregate}(\mathcal{Y}_i, R_i)$
  `// Batch loss over (non-empty) series present in` $\mathcal{W}_0$
  $I \leftarrow \{\, i \in \{1, \ldots, n\} \mid \mathcal{W}_i \neq \varnothing \,\}$
  $\mathcal{L}_{\text{batch}} \leftarrow \dfrac{1}{|I|} \displaystyle\sum_{i \in I} \mathcal{L}\big(\hat{y}^{(i)},\, y^{(i)}\big)$
  `// Parameter update`
  $\theta \leftarrow \textsc{Update}(\theta,\, \mathcal{L}_{\text{batch}})$
  `// Remove sampled windows from the pool`
  $\mathcal{W} \leftarrow \mathcal{W} \setminus \mathcal{W}_0$
return θ

Algorithm 2: AGGREGATE

Input : Windows for series i: $\mathcal{W}_i$;
$\mathcal{Y}_i \leftarrow \{\, f_\theta(w) \mid w \in \mathcal{W}_i \,\}$ `// Window-level predictions`
Retrieval map R_i with $R_i[k] = \big(s_k^{(1)}, \ldots, s_k^{(m)}\big)$ `// top-`m
Temperature $\tau > 0$

Output : Series-level probability $\hat{y}^{(i)} \in \mathbb{R}^C$

`// 1) Summarize neighbor support per window`
foreach $k \in \mathcal{W}_i$ **do**
  $\bar{s}_k \leftarrow \dfrac{1}{m} \sum_{j=1}^{m} s_k^{(j)}$ `// mean similarity for window` k

`// 2) Softmax weights across windows`
foreach $k \in \mathcal{W}_i$ **do**
  $a_k \leftarrow \exp\big(\bar{s}_k / \tau\big)$
$Z \leftarrow \sum_{t \in \mathcal{W}_i} a_t$
foreach $k \in \mathcal{W}_i$ **do**
  $\alpha_k \leftarrow a_k / Z$ `//` $\alpha_k \geq 0$, $\sum_k \alpha_k = 1$

`// 3) Aggregate in probability space`
$\hat{p}^{(i)} \leftarrow \sum_{k \in \mathcal{W}_i} \alpha_k \, \mathcal{Y}_{ik}$

return $\hat{y}^{(i)}$

B Dataset and Preprocessing

B.1 Dataset

Following the protocol of [9], we use a multicenter iEEG cohort with clinical annotations of the seizure onset zone (SOZ). For each site, we report the number of patients recorded (n), the number with SOZ labels (n_{SOZ}), the total number of channel time series (n_{ts}), the proportion of SOZ labeled (p_{SOZ}), the iEEG modality, nominal sampling frequency, and availability of postoperative outcome labels. A summary is provided in Table 2.

Table 2: Multicenter iEEG summary. n: patients recorded; n_{SOZ}: patients with SOZ annotation; n_{ts}: channel time series; p_{SOZ}: fraction of series labeled SOZ.

Medical Center	n	$n_{\textbf{SOZ}}$	$n_{\textbf{ts}}$	$p_{\textbf{SOZ}}$	**iEEG Type**	**Freq (Hz)**	**Outcomes**
JHH	7	3	1458	7.48%	ECoG	1000	No
NIH	14	11	3057	12.23%	ECoG	1000	Yes
UMMC	9	9	2967	5.56%	ECoG	250–1000	Yes
UMF	5	1	129	25.58%	ECoG	1000	No

Per Mootoo et al. [9], we filter to patients with SOZ annotations when forming the supervised subsets (n_{SOZ}). Because SOZ vs. non-SOZ is highly imbalanced at the series level, later class balancing reduces the effective number of training/validation examples for each site.

B.2 Data preprocessing

Unless stated otherwise, we largely adhere to [9]. Each patient contributes multiple channels (electrodes). For every site we:

1. Extract all channels and form per-channel univariate time series;
2. Perform class balancing so that SOZ and non-SOZ series counts are equal within the training/validation splits (non-SOZ downsampling);
3. Split channels into train/validation/test at approximately 70% /10% /20%, ensuring no temporal leakage across splits during window sampling;
4. Z-score normalize each channel independently to zero mean and unit variance.

We report F1, AUC, and accuracy in the main results.

B.3 Reproducibility & Hyperparameters

All training hyperparameters are listed in Table 3. Each experiment is run with five fixed seeds (69421–69425). We will release the full codebase, configuration files, and run scripts in a public repository at camera-ready, including exact commands and environment specifications to reproduce Table 1.

B.4 Computational Resources

Experiments were conducted on a single NVIDIA T4 GPU with 32 GB system RAM, each training run (per seed) took about 1 hour. All computations used PyTorch with CUDA [26].

Table 3: RAxSS hyperparameters and data settings.

Experiment / Reporting	
Model ID	`PatchTSTBlind`
Seeds	$[69421, 69422, 69423, 69424, 69425]$
Learning type	`sl` (supervised)
Metrics & selection	report `acc`, `ch_acc`, others; tune on `ch_f1`; select on `ch_acc`
Task	`classification`
GPU	`gpu_id=0`; single-GPU runs (see B.4)
Data / Sampling / Preprocess	
Dataset	`open_neuro` (multicenter iEEG)
Split	train/val/test = 0.7/0.1/0.2, *no temporal leakage*; class balancing in train/val
Windowing	length L=1024, stride = 5; univariate channels (C=1)
Batching	length-proportional stochastic sparse sampling (SSS)
Resizing	`pad_trunc`; `seq_load=True`; `num_workers=8`
Scaling	per-channel z-score; `scale=True`; `shuffle_test=True`
Backbone / Architecture	
Encoder layers	`num_enc_layers=2`
Dims / heads	`d_model=32`, `d_ff=128`, `num_heads=4`
Dropout	`attn_dropout=0.3`, `ff_dropout=0.3`, `pred_dropout=0.0`
Head	`linear`
RevIN	`revin=True`, `revin_affine=True`, `revout=False`
Retrieval & Aggregation (RAxSS)	
Similarity	Pearson or cosine (within series/channel)
Support $\rightarrow$ weights	average top-m [10] similarities; softmax with temperature $\tau > 0$ across windows
Aggregation	In probability space (Alg. 2)
Use relevance	`use_relevance=True`
Optimization / Training	
Epochs & batch	`epochs=50`, `batch_size=8192`
Optimizer	`adam`, `weight_decay=1e-6`
Scheduler	cosine warmup: `warmup_steps=100`, `T_max=700`, `start_lr=0.0`, `final_lr=1e-6`, `max_lr=3e-4`
Early stopping	`patience=5`
Loss	`BCE` (`ch_loss=True`, type `BCE`, α=0.0, β=1.0)
Dataset-specific (OpenNeuro settings)	
Kernels/pooling	`kernel_size=24`, `kernel_stride=-1`, `pool_type=avg`
Centers	`all_clusters=True`
Task	binary classification (`pred_len=1`)