UAB
Universitat Autònoma de Barcelona

Dipòsit digital de documents de la UAB

This is the **published version** of the bachelor thesis:

Cases Font, Marc. *A Comparative Study of Feature Extractors for Epileptic Seizure Detection*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Grau en Enginyeria de Dades)

This version is available at https://ddd.uab.cat/record/337356

under the terms of the  license.

A Comparative Study of Feature Extractors for Epileptic Seizure Detection

Marc Cases Font
Bachelor's Degree in Data Engineering
Universitat Autònoma de Barcelona (UAB)
Centre de Visió per Computador (CVC)
Marc.CasesF@autonoma.cat

***Abstract*—Epilepsy affects approximately 50 million people worldwide, and roughly one third of patients continue to experience seizures despite medication. Clinical monitoring by means of electroencephalogram produces continuous EEG recordings spanning hours or days, within which the relevant ictal events may last only seconds, and their identification still relies largely on visual inspection by a specialist—a slow, costly and error-prone process that does not scale to the volume of data generated in practice. Automatic seizure detection aims to close this gap. This work isolates a single design question within that goal: how the transformation of the raw EEG signal into features affects the detection performance of a deep learning model. Rather than proposing a complete new architecture, we fix the downstream classifier and the training pipeline and vary only the feature-extraction stage. Six variants are defined along two orthogonal axes—per-channel compression and inter-channel mixing—of which four are evaluated on the public CHB-MIT scalp-EEG database. To assess the reliability of the reported results, each variant is tested under three validation protocols of increasing strictness, culminating in a fully patient-independent scheme in which the model is evaluated only on subjects never seen during training. We report sensitivity, precision, specificity and F1, prioritizing sensitivity as the clinically most relevant metric, since a missed seizure is the costliest type of error.**

***Index Terms*—Epileptic seizure detection, *deep learning*, EEG, feature extraction, CNN, LSTM, Transformer, autoencoder, k-fold cross-validation, leave-subjects-out.**

I. Introduction

Epilepsy is one of the most common chronic neurological disorders, affecting roughly 50 million people around the world [2]. What defines it is the seizure: a sudden burst of abnormal electrical activity in the brain that can cause anything from a brief lapse of awareness to a full convulsion with loss of consciousness. Seizures are hard to predict, and that unpredictability is a large part of what makes epilepsy so disruptive to live with. For about a third of patients, seizures continue even with medication, and for them the ability to monitor brain activity reliably and continuously is what allows clinicians to understand the disorder, adjust treatment and keep the patient safe.

Brain activity is recorded with electroencephalography (EEG), a set of electrodes placed on the scalp. The catch is one of scale: monitoring a patient may mean recording for hours or even days, and within all that data a real seizure might last only seconds. Those few seconds sit buried in long stretches of ordinary activity, and today finding them mostly comes down to a specialist reading the recording by eye. That is slow, costly and easy to get wrong, and it simply does not scale to the amount of data a hospital produces. This is the gap that *automatic seizure detection* tries to close, and it is what makes the problem both practically useful and worth studying.

Building such a system is harder than just handling long, noisy recordings. Two difficulties stand out. First, seizures are rare compared with normal activity, so a detector has to cope with a heavy imbalance between the two. Second, and more troubling, seizures look quite different from one person to the next: a model that does well on patients it has already seen can fail completely on someone new. This difficulty of generalizing across people is, in practice, the main obstacle that automatic detection has to overcome.

Whatever the system, it has to turn the raw recording into a compact representation that keeps what matters for diagnosis and discards the rest. How that representation is built can strongly affect performance, yet most published work presents one complete system end to end, which makes it hard to tell which design decisions are actually doing the work. This study takes a different route. It frames the problem as a *controlled comparison*: every part of the processing chain is held fixed, and only the feature-extraction step changes. That way, any difference in performance can be traced back to that single choice. Six variants are considered, four of which are evaluated here.

A second concern is whether the evaluation itself can be trusted. It is well known that results in this field can be inflated when recordings from the same patient end up in both training and testing. To measure how much this matters, each variant is assessed under three validation protocols of increasing strictness, ranging from a permissive scheme that freely mixes data from the same patient across splits to a fully patient-independent one in which the model is tested only on individuals it has never seen. All experiments use the publicly available CHB-MIT scalp-EEG database [3], [15].

The main contributions of this work are:

1) A complete and reproducible processing pipeline for continuous scalp EEG recordings that prepares the CHB-MIT database for automatic seizure detection, including signal cleaning, segmentation and labeling.
2) A family of six feature-extraction variants organized along two interpretable dimensions—how each electrode channel is individually summarized and how the infor-

mation from all channels is then combined—of which four are fully implemented and evaluated.
3) A controlled comparison that directly measures the contribution of each design choice by pairing variants that differ in exactly one dimension.
4) A systematic analysis of how the choice of validation protocol affects reported performance, with results presented both overall and per patient, prioritizing sensitivity as the clinically most relevant metric.

II. Related work and state of the art

The automatic analysis of seizures from EEG has progressively moved from hand-crafted descriptors to *deep learning* methods, which learn the representations directly from the signal [13]. In particular, convolutional neural networks have been widely used for epileptic seizure prediction and detection, both in intracranial and scalp EEG, because they can extract local temporal patterns from raw or lightly preprocessed signals without requiring manual *feature* engineering [8]–[10]. This motivates the use of convolutional modules as supervised temporal extractors, especially when the goal is to learn short-term EEG patterns directly from the signal.

A recent study on across-subject performance in EEG analysis is particularly relevant to this project [1]. That work uses seizure detection on the CHB-MIT database as a case study and shows that validation at the window level can produce over-optimistic results when samples from the same subject or state appear in both training and test partitions. To address this issue, it evaluates models at increasingly strict hierarchical levels: window, state and subject. This motivates the validation philosophy adopted here, where the same experimental pipeline is evaluated under progressively stricter partition schemes in order to measure how reported performance changes as the independence between training and validation increases.

The same study also compares several neural architectures for multi-channel EEG fusion, including early and mid-level concatenation, trainable weighted averages, multiple weighted averages and an LSTM-based model [1]. This is relevant because automatic EEG analysis does not only depend on the temporal features extracted from each signal, but also on how information from multiple electrodes is combined. Fixed concatenation preserves channel-specific information but leaves the classifier to learn the relations between channels, whereas learnable fusion mechanisms can assign different importance to different parts of the multichannel representation. This work follows the same general motivation of comparing alternative ways of representing multi-channel EEG, but focuses on isolating the effect of the window-level *feature* extractor while keeping the downstream classifier fixed.

Temporal modelling is another important aspect of EEG analysis. Since EEG is a time series, recurrent networks such as LSTMs are well suited to capture dependencies over time [7]. In many previous works, the recurrent model is part of a complete end-to-end classifier. In contrast, the approach followed here separates window-level representation from temporal decision modelling: each 2-second EEG *window* is first transformed into an embedding, and a shared LSTM then models the evolution of consecutive embeddings over a longer decision *chunk*. This design makes it possible to compare different window representations under the same temporal classifier.

Finally, attention mechanisms and Transformers have shown that treating the input units as a set of *tokens*—for example, the *patches* of an image in ViT [5]—enables a flexible, content-adaptive aggregation of information. In the context of EEG, this idea can be transferred to the channel dimension by treating each electrode channel, or each channel descriptor, as a token to be mixed by a Transformer *encoder*. A complementary line of research explores *self-supervised* EEG representation learning, where models are trained without labels, for example through reconstruction, and the resulting *encoder* is reused in downstream tasks. Works such as VAEEG [14] support this idea in settings where labels are scarce, which is a common situation in seizure detection. Overall, the present work brings these ideas together in a controlled comparison: the temporal classifier and training protocol are kept fixed, and only the window-level *feature* extraction stage is varied.

III. Problem formulation

A. *SPH/SOP framework and transition* buffers

The temporal regions around each seizure are parameterized by means of two quantities taken from the seizure-prediction literature [4]: the *Seizure Prediction Horizon* (SPH) and the *Seizure Occurrence Period* (SOP). Here they are not used to anticipate the seizure, but only to define the preictal *buffer* that is removed from the negative class.

Let T_0 be the seizure onset instant. The preictal interval associated with each seizure is

$$\mathcal{I}_{\text{pre}} = [\, T_0 - (\text{SPH} + \text{SOP}),\ T_0 - \text{SPH}\,]. \tag{1}$$

In this work SPH $= 0$ min and SOP $= 5$ min, so that the preictal *buffer* spans the five minutes immediately preceding the onset. A symmetric *postictal buffer* of 5 min is defined after the end of the seizure. Both *buffers* are excluded from the classification targets: this prevents the ambiguous transition activity adjacent to a seizure from being labeled as interictal, keeping the two classes well separated.

B. *Temporal states and labeling*

Each temporal instant of the signal is classified into one of the following four states:

- **Ictal**: during the seizure ($[T_0^{\text{start}}, T_0^{\text{end}}]$).
- **Postictal**: recovery period after the seizure (5 min).
- **Preictal**: the interval $\mathcal{I}_{\text{pre}}$ defined above (5 min before the seizure).
- **Interictal**: the remaining time, far from any seizure.

The classification task addressed in this work is binary: **ictal (1) vs. interictal (0)**. The preictal and postictal *windows* are not used as classification targets (they are neither the positive nor the negative class), **but they may appear in the context *chunk* of a sample.**

Table I: Selected EEG channels (22 channels).

FP1-F7	F7-T7	T7-P7	P7-O1
FP1-F3	F3-C3	C3-P3	P3-O1
FP2-F4	F4-C4	C4-P4	P4-O2
FP2-F8	F8-T8	T8-P8	P8-O2
FZ-CZ	CZ-PZ	P7-T7	T7-FT9
FT9-FT10	FT10-T8		

IV. Dataset: CHB-MIT

The CHB-MIT dataset [3], [15], distributed through PhysioNet [12], contains continuous scalp-EEG recordings of 24 pediatric subjects with epilepsy, recorded at the Children's Hospital Boston. The recordings are in EDF format at 256 Hz with a bipolar montage. The dataset includes a total of 198 annotated seizures. To guarantee consistency across subjects, 22 EEG channels are selected (Table I); the non-EEG channels (ECG, VNS) and the *dummy* channels are discarded.

A. *Decision* windows *and* chunks

The continuous EEG signal is divided into *windows* of duration $L = 2$ s without overlap. Each *window* contains $C = 22$ channels with $L \cdot f_s = 512$ samples per channel (at 256 Hz), forming a tensor $X_i \in \mathbb{R}^{22\times512}$. The 2 s duration is a compromise: it is short enough to temporally localize the seizure onset within the *chunk*, while at the same time providing enough samples (512) for the per-channel extractor to capture meaningful temporal patterns; it is also a common choice in the EEG literature.

A decision *chunk* groups $N = 30$ consecutive *windows* (equivalent to 60 s of context). This length is another compromise: 60 s offer enough temporal context for the LSTM to model the evolution of the signal dynamics around the seizure, while keeping the sequence in a regime in which the LSTM trains reliably with the available compute and data budget. The label of a *chunk* is the label of its *last window*, and a *chunk* is admitted into the dataset only if its last *window* is **ictal or interictal**. The *chunk stride* depends on the class of that last *window*: ictal *chunks* are extracted with *stride* 1 (consecutive ictal *chunks* share 29 of their 30 *windows*), which densely samples the scarce positive class, whereas interictal *chunks* are extracted with a *stride* equal to the *chunk* length ($N = 30$ *windows*), so that interictal *chunks* do *not* overlap. This asymmetry preserves every available ictal *chunk* while removing the heavy redundancy between near-identical interictal *chunks* and reducing their number by roughly $30\times$. The 29 previous context *windows* may belong to any phase, including preictal or postictal: this preserves the realistic continuity of the EEG signal in the moments immediately before the target window.

V. Signal preprocessing

The preprocessing *pipeline* is applied to each EDF file and consists of the following steps:

a) Channel selection: A correspondence is built between the channel names of the EDF file (which may contain numeric suffixes or spaces) and the 22 canonical channels. Each name is cleaned by removing the reference suffixes (e.g. T8-P8-0 $\rightarrow$ T8-P8). If any canonical channel is missing, the file is excluded.

b) Normalization: A per-channel, per-recording *z-score* normalization is applied:

$$\hat{x}_c(t) = \frac{x_c(t) - \mu_c}{\sigma_c + \epsilon}, \quad \epsilon = 10^{-8}, \tag{2}$$

where μ_c and σ_c are the mean and standard deviation of channel c over the whole recording. The normalization is applied *before* windowing, so that all *windows* of the same recording share the same statistics.

c) Segmentation into windows*:* The normalized signal is divided into consecutive *windows* of $L = 2$ s, obtaining a tensor $X_i \in \mathbb{R}^{22\times512}$ per *window*.

VI. System architecture

A. *Overview and shared* pipeline

The strategies share the same global *pipeline* (Fig. 1). A function $g(\cdot)$ transforms each *window* $X_i \in \mathbb{R}^{22\times512}$ into a *feature* vector $v_i \in \mathbb{R}^{d_{\text{window}}}$. A sequence of $N = 30$ consecutive vectors $[v_{i-29}, \ldots, v_i]$ is then fed into a shared LSTM-based classifier that produces the ictal probability of the last *window*:

$$p_i = \sigma\big(\text{Head}\big(\text{LSTM}([v_{i-29}, \ldots, v_i])\big)\big) \in (0, 1). \tag{3}$$

The only element that differs between the six strategies is $g(\cdot)$. Everything downstream of g (sequence assembly, LSTM, head, *loss*, optimizer, *schedule*, evaluation) is identical. Fig. 1 shows this shared skeleton with $g(\cdot)$ drawn as a single generic block; the six concrete extractors that can occupy that block are described one by one in Section VI-C and drawn together in Fig. 2 (Appendix A).

B. *Shared classifier and head*

The shared classifier is a two-layer LSTM [7] with the following configuration:

- input_size = d_{window} (varies depending on the strategy).
- hidden_size = 256.
- num_layers = 2.
- dropout = 0.3 between layers.

Given the input sequence $[v_{i-29}, \ldots, v_i]$, the LSTM produces a sequence of *hidden states*. The *hidden state* of the last layer at the last time step, $h_T \in \mathbb{R}^{256}$, is taken as the summary of the *chunk* and is passed to the classification head:

$$\text{Head}(h_T) = \text{Linear}_{256\to1}\big(\text{Dropout}(\text{LayerNorm}(h_T))\big), \tag{4}$$

which produces a single *logit*. The sigmoid of this *logit* is interpreted as the probability that the last *window* of the *chunk* is ictal.

C. *Feature-extraction strategies*

The six strategies explore two orthogonal axes (Table III):

- **Per-channel compression**: none (raw signal), supervised (CNN trained *end-to-end*) or self-supervised (autoencoder pretrained and then *fine-tuned*).
- **Inter-channel mixing**: none (*flatten* over the channels) or attention-based (Transformer *encoder* with CLS *token*).

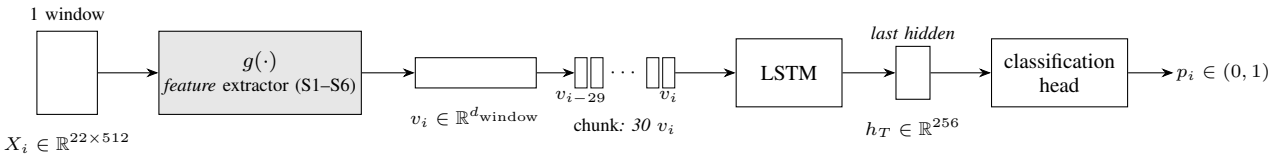


Figure 1: Shared system *pipeline*. A *feature* extractor $g(\cdot)$ maps each 2-second *window* $X_i \in \mathbb{R}^{22\times512}$ to a *feature* vector $v_i \in \mathbb{R}^{d_{\text{window}}}$. A *chunk* of $N = 30$ consecutive vectors $[v_{i-29}, \ldots, v_i]$ is fed to a shared two-layer LSTM, whose last *hidden state* $h_T \in \mathbb{R}^{256}$ is passed to the classification head to produce p_i, the probability that the *last window* of the *chunk* is ictal. Everything downstream of $g(\cdot)$ is identical across the six case studies; only the shaded block $g(\cdot)$ varies (Fig.~2).

Table II: Shared classifier and training configuration across the strategies.

Element	Value
LSTM *hidden* / layers	256 / 2
LSTM *dropout*	0.3
Head	LayerNorm → Dropout → Linear(256 → 1)
Loss	BCEWithLogitsLoss
Optimizer	Adam
Grad clipping	1.0
Scheduler	None (constant learning rate)
Epochs	30
Cross-validation	k-fold ($K = 5$): chunk / recording / patient
Model selection	last-epoch metrics (30)

Table III: The six strategies along two orthogonal axes. S1–S4 are evaluated in this work; S5–S6 are deferred.

	Flatten	**Transformer + CLS**
No compressor	S1 (11 264)	S3 (128)
Supervised CNN	S2 (1 408)	S4 (128)
Self-supervised AE	S5 (2 816)†	S6 (128)†

The values in parentheses are d_{window}. †Deferred (not evaluated in this iteration).

The six extractors are drawn together in Fig. 2 (Appendix A), each truncated at the *feature* vector v_i that feeds the shared *pipeline* of Fig. 1. In all cases, identical $(22, 512)$ *windows* enter g and a sequence of 30 vectors is subsequently delivered to the shared LSTM described in Section VI-B.

D. Strategy 1 (S1): *Raw-signal* baseline *with* Flatten

S1 is the lower reference of the study. It completely removes the *feature* extractor and feeds the LSTM directly with the raw signal of the *window*, channel by channel, without any intermediate inductive bias. Its role is to quantify how much the LSTM can solve on its own and to provide the fair *baseline* against which the gains of the other strategies are measured.

The extractor g is the identity followed by a *flatten*. Given a *window* $X_i \in \mathbb{R}^{22\times512}$, the 22 channels are concatenated in canonical order to form a single vector:

$$v_i = \left[x_{i,1}^{(1)}, \ldots, x_{i,1}^{(512)}, \ \ldots, \ x_{i,22}^{(512)}\right]^\top \in \mathbb{R}^{11\,264}. \qquad (5)$$

The channel identity is encoded *positionally*: the first 512 entries always correspond to FP1-F7, the next 512 to F7-T7, and so on. The model must learn this implicit structure from the data, without the help of any explicit channel *embedding*.

There is no pretraining: the LSTM and the head are trained *end-to-end* with the classification *loss*.

Table IV: Per-channel CNN used in S2 and S4. Input: one channel of 512 samples; output: 64 dimensions.

Block	**Channels**	**Kernel**	**Pool**
Conv1d + BN + GELU	$1 \to 16$	7	MaxPool(4)
Conv1d + BN + GELU	$16 \to 32$	5	MaxPool(2)
Conv1d + BN + GELU	$32 \to 64$	3	—
Temporal summary	64	—	—
Dropout(0.1) + Linear	$64 \to 64$	—	—

E. Strategy 2 (S2): *Per-channel CNN with shared weights* + Flatten

S2 introduces a per-channel *feature* extractor while keeping the rest of the *pipeline* as similar as possible to S1. A small 1D CNN with weights shared across the 22 channels learns generic local temporal patterns, transforming each channel into a compact descriptor before concatenating them into v_i.

The CNN consists of three convolutional blocks followed by a global *pooling* and a linear projection (Table IV). The 22 channels of the *window* are processed in parallel with this same network, producing 22 descriptors $f_{i,c} \in \mathbb{R}^{64}$ that are concatenated in canonical order:

$$v_i = [f_{i,1};\ f_{i,2};\ \ldots;\ f_{i,22}] \in \mathbb{R}^{22\times64} = \mathbb{R}^{1\,408}. \qquad (6)$$

After the convolutional blocks, the temporal information of each channel is summarized into a fixed-size descriptor of 64 values. Therefore, each descriptor represents the whole 2-second channel segment rather than a shorter sequence of temporal samples. The CNN is responsible for learning which local temporal patterns are relevant within each individual channel, while the LSTM models how the resulting window-level representations evolve across consecutive *windows*.

With $d_{\text{window}} = 22 \times 64 = 1,408$ (an $8\times$ reduction with respect to S1), the CNN, the LSTM and the head are trained jointly *end-to-end* with the classification *loss*, without any prior pretraining.

F. Strategy 3 (S3): *Inter-channel Transformer over the raw signal*

S3 changes the inter-channel fusion mechanism: instead of concatenating and letting the LSTM discover the relations, a Transformer *encoder* [6] is applied within each *window* that treats the channels as *tokens* and mixes them via *self-attention*. It also checks whether the inter-channel mixing can be performed directly on the raw signal, without prior per-channel *feature* extraction.

Each channel of the *window* is treated as a *token* whose initial *embedding* is its temporal vector of 512 samples. A learnable CLS *token* $f_{\text{CLS}} \in \mathbb{R}^{512}$ is prepended:

$$S_i^{(0)} = \left[f_{\text{CLS}};\, x_{i,1};\, \ldots;\, x_{i,22}\right] \in \mathbb{R}^{23\times 512}. \tag{7}$$

Each of the 23 *tokens* is linearly projected to $d_{\text{model}} = 128$ via Linear$(512 \to 128)$, and the resulting sequence is passed through a Transformer *encoder* with the following configuration: 4 *pre-norm* layers, 4 *heads*, *feedforward* dimension 512 and GELU activation. **No channel-identity *embedding* is added**: the canonical order of the list does not encode real topological information, so the channels are treated as an unordered set. The *window* representation is the output at the CLS *token*: $v_i = h_{\text{CLS}}^{(L)} \in \mathbb{R}^{128}$.

With $d_{\text{window}} = 128$ (the smallest of the study, shared with S4 and S6), the cost of the attention over 23 *tokens* is negligible; the dominant cost comes from the Linear$(512 \to 128)$ projection, which the Transformer must learn from scratch together with the attention and the CLS summary. Transformer, LSTM and head are trained *end-to-end* without pretraining.

G. Strategy 4 (S4): Per-channel CNN + inter-channel Transformer

S4 combines the ideas of S2 and S3: first a per-channel CNN compresses each channel into a low-dimensional descriptor, and then a Transformer mixes the information across the resulting descriptors. It is the "light and filtered" version of S3, aligned with the typical inductive biases in EEG modeling: CNN for the local *features*, attention for the inter-channel relations.

The extractor has two stages. The first is a per-channel CNN identical to that of S2 (Table IV, shared weights), which produces 22 descriptors of 64 dimensions, $F_i \in \mathbb{R}^{22\times 64}$. The second is an inter-channel Transformer: the 22 descriptors are treated as *tokens*, a CLS *token* of dimension 64 is prepended and the 23 *tokens* are projected to $d_{\text{model}} = 128$ via Linear$(64 \to 128)$. Then the same Transformer *encoder* as in S3 is applied (without channel *embedding*). The output at the CLS *token* gives $v_i = h_{\text{CLS}}^{(L)} \in \mathbb{R}^{128}$.

With respect to S3, the Transformer operates on more compact and, in principle, more informative *tokens*: it no longer has to learn the temporal *feature* extraction implicitly and can devote its capacity to the inter-channel relations.

With $d_{\text{window}} = 128$, the CNN, the Transformer, the LSTM and the head are trained jointly *end-to-end* with the classification *loss*. There is no pretraining; the CNN is initialized randomly, as in S2.

H. Strategy 5 (S5): Pretrained per-channel autoencoder + Flatten *(with* Fine-Tune*)*

Note: S5 is described for completeness but is deferred to a later iteration; it is not part of the experimental evaluation in Section VIII.

S5 replaces the supervised CNN of S2 with an *encoder* pretrained in a self-supervised manner to reconstruct EEG signals. The motivation is the scarcity of the ictal label: with 24 patients, the seconds of seizure are few compared with the total signal available. Pretraining a residual convolutional autoencoder [11] that operates channel by channel, over all the *windows* of the corpus (in all their phases and without labels), allows the *encoder* to learn a general representation of the signal that is then specialized via *fine-tuning* on the classification task.

After pretraining, the *decoder* is discarded and the *encoder* is applied per channel, producing 22 *embeddings* that are concatenated in canonical order into a vector $v_i \in \mathbb{R}^{22\times 128} = \mathbb{R}^{2\,816}$ (the channel identity is encoded positionally, as in S1 and S2). With $d_{\text{window}} = 2\,816$, intermediate between S1 and S2, S5 occupies the *flatten* cell of the self-supervised row of the 2×3 design. The details of the *encoder* architecture and of the *fine-tuning* scheme will be specified when this strategy is implemented and evaluated.

I. Strategy 6 (S6): Pretrained per-channel autoencoder + Transformer over the F_i *(with* Fine-Tune*)*

Note: S6 is described for completeness but is deferred to a later iteration; it is not part of the experimental evaluation in Section VIII.

S6 combines the two most powerful choices of the study: the self-supervised *features* of S5 with the Transformer-based inter-channel mixer of S3 and S4. The hypothesis is that they are complementary: the pretrained *encoder* provides per-channel descriptors that have seen far more signal than any supervised CNN on this dataset, and the Transformer replaces the rigid positional concatenation of S5 with a content-adaptive aggregation.

Over the 22 *embeddings* of the *encoder* ($F_i \in \mathbb{R}^{22\times 128}$) the same inter-channel Transformer as in S3 and S4 is applied (with CLS *token*, without channel *embedding*), so that $v_i = h_{\text{CLS}}^{(L)} \in \mathbb{R}^{128}$. S6 is to S5 what S4 is to S2: it adds an inter-channel mixer over per-channel *features*; the difference with S4 lies in the origin of those *features* (supervised from scratch in S4, self-supervised in S6). With $d_{\text{window}} = 128$, S6 completes the lower-right cell of the 2×3 design and enables the supervised vs. self-supervised comparison with a mixer (S4 vs. S6). As with S5, it is deferred to the next iteration.

J. Comparative analysis: what each pair isolates

The strategies are organized so that each axis of variation can be examined in isolation by comparing the corresponding pairs (Table III). In this iteration only the pairs within S1–S4 are quantified.

a) Per-channel compression: The vertical axis S1 → S2 (without mixer) and S3 → S4 (with mixer) measures the contribution of adding a per-channel *feature* extractor.

b) Inter-channel mixing: The horizontal axis S1 → S3 and S2 → S4 measures the contribution of the Transformer mixer versus *flatten* + LSTM at each compression level.

The comparisons involving the self-supervised *encoder* (S2 vs. S5, S4 vs. S6, S5 vs. S6) are deferred to the next iteration. This structure makes it possible to attribute the performance gains or losses to specific components rather than to the architecture as a whole.

VII. Experimental setup

A. Partition (k-fold) schemes and class balancing

Validation is performed with **three k-fold cross-validation schemes** ($K = 5$ in all three) that differ only in the partition's *grouping unit*. Each grouping unit defines a different *generalization level*: from generalization to held-out *chunks* drawn from the same recordings and patients, to generalization to unseen *recordings*, and finally to generalization to unseen patients. The schemes are ordered from least to most strict, that is, from the highest to the lowest risk of information leakage between training and validation. Comparing the three makes it possible to quantify how much performance is inflated when the separation criterion is relaxed, and how much each strategy degrades as the required *generalization level* becomes more demanding.

a) Partition by chunk*:* The valid *chunks* are shuffled with a fixed seed and distributed evenly across $K = 5$ *folds*, without regard to the patient or *recording* of origin; each *fold* holds out 20 % of the *chunks* for validation (80/20). It is the most optimistic scheme: since consecutive ictal *chunks* share 29 of their 30 *windows* (*stride* 1), nearly identical ictal *chunks* may end up split between training and validation, introducing strong temporal leakage on the positive class (the interictal *chunks*, sampled without overlap, do not contribute to this leakage). It acts as an optimistic upper bound on performance.

b) Partition by recording*:* The grouping unit is the EDF file. The valid *recordings* are shuffled with a fixed seed and distributed evenly across the $K = 5$ *folds*, without regard to the patient of origin; for each *fold* that group of *recordings* is the validation set and the rest are the training set. No *recording* is split between training and validation, so the temporal overlap *within* a recording can no longer leak between the two sets. The split is *not* stratified by patient: which subjects fall in the validation set of a given *fold* depends only on where their *recordings* land, so a patient is not guaranteed to appear in every *fold*. Because most patients have several *recordings* spread across *folds*, inter-*recording* leakage per subject still persists: the scheme measures generalization to *new recordings of already-seen patients* and is the one of intermediate strictness, between the by-*chunk* and the *leave-subjects-out* schemes.

*c) Partition by patient (*leave-subjects-out*):* The grouping unit is the subject. The 24 patients are shuffled with a fixed seed and distributed evenly across $K = 5$ disjoint groups of approximately equal size. For each *fold* $k \in \{1, \dots, K\}$, the patients in group k form the validation set and the remainder form the training set. Since the partition is at the subject level, no patient ever appears simultaneously in training and validation, which forces evaluation against unseen subjects—the regime relevant to clinical generalization and the most demanding of the three.

In all three schemes, the entire *pipeline* (data construction, model initialization and training) is repeated independently for each of the five *folds*.

Within each *fold*, once all valid *chunks* have been collected, class balancing is applied **only to the training set**: its interictal *chunks* are randomly subsampled (with a fixed seed, for reproducibility) to achieve a 1:1 ratio against the ictal ones, so that the supervised gradient signal remains balanced between the two classes. The **validation set is not re-balanced to 1:1**: all ictal *chunks* and all (non-overlapping) interictal *chunks* are kept. Because the interictal *chunks* are sampled without overlap (*stride* N) while the ictal ones are sampled with *stride* 1, the validation set retains a realistic class imbalance—interictal still outnumbers ictal—without either the artificial 1:1 balancing of the training set or the strong redundancy that overlapping interictal *chunks* would introduce. The reported *precision* and *specificity* are thus measured on a set that keeps a substantial interictal majority, a more conservative estimate than a balanced set would give, while remaining tractable in memory and compute.

B. Hyperparameters

Table II provides a summary of the hyperparameters fixed across the strategies. The specific hyperparameters of the Transformer used in S3, S4 (and S6) are detailed in Section VI-F. The autoencoder used in S5 and S6 is described in Section VI-H.

C. Evaluation protocol and metrics

Each strategy is trained and validated under the **three k-fold partition schemes** described above (Section VII-A; $K = 5$ *folds* each: by *chunk*, by *recording* and by patient). The reported metrics are those of the **last training epoch (epoch 30)**, not those of the best *checkpoint* according to validation F1. Reporting the last epoch avoids selecting *a posteriori* the epoch with the best validation and more realistically reflects the performance to be expected at deployment, where it is not known in advance which epoch will turn out to be optimal. At the *window* level we report *recall* (sensitivity), *precision*, *specificity* and the F1 score, computed from the validation confusion matrix as

$$\begin{aligned} \text{recall} &= \tfrac{TP}{TP+FN}, & \text{precision} &= \tfrac{TP}{TP+FP}, \\ \text{specificity} &= \tfrac{TN}{TN+FP}, & \text{F1} &= \tfrac{2PR}{P+R}. \end{aligned} \tag{8}$$

All metrics are aggregated as the **mean ± standard deviation across the** $K = 5$ ***folds*** of each scheme, which reflects the variability of the performance across the different partitions. Because the validation set is not re-balanced to 1:1 (Section VII-A), *precision* and *specificity* are measured under a realistic class imbalance rather than on an artificially balanced set.

In addition to this *fold*-level aggregation, for each partition scheme we also report **per-patient** metrics: by *patient* (Tables XIa and XIb), by *chunk* (Tables IXa and IXb) and by *recording* (Tables Xa and Xb). In every scheme the per-patient confusion matrix covers each subject's validation *chunks* exactly once, but it is assembled differently: in the *leave-subjects-out* scheme the subject is held out in a single *fold*, so its matrix comes from that *fold*; in the by-*chunk* and by-*recording* schemes the subject's validation units (its *chunks*, respectively its *recordings*) are distributed across the *folds*, and since each unit is held out in exactly one *fold*

the per-patient confusion matrix is obtained by *accumulating* the held-out counts of that subject across the five *folds*. In all three cases the four metrics are computed individually per patient—always at the last epoch—which exposes the strong inter-patient variability that the *fold*-level average hides and identifies the subjects on which the detector fails. The per-patient values are summarized as the mean $\pm$ standard deviation across the 24 patients; metrics that are undefined for a given patient (e.g. *precision* when the model issues no positive prediction for that subject) are excluded from this average.

In this task the cost profile is asymmetric: a false negative (failing to detect a seizure) is considered more serious than a false positive, so the clinically prioritized metric is *recall* (sensitivity), which measures the fraction of detected seizures. Nevertheless, as a single-number summary to compare the strategies, F1 is used, which combines *recall* and *precision* and avoids rewarding models with high *recall* at the cost of a tiny *precision*. Both metrics are reported, and *recall* is interpreted as the main indicator of clinical safety.

VIII. Results and discussion

This section presents the validation results of the four strategies evaluated in this iteration (S1–S4). The analysis focuses on three questions: how much the validation protocol affects the reported performance, which feature-extraction components improve generalization, and how stable the best model is across patients. Throughout the section, *recall* is interpreted as the clinically most relevant metric, whereas F1 is used as the main single-number summary for comparing strategies.

A. Overall performance and validation leakage

Tables V–VII report the validation metrics under the three partition schemes, ordered from the least strict protocol, by *chunk*, to the most strict one, by patient. All values are computed at the *window* level in the last training epoch and are reported as the mean $\pm$ standard deviation across the five *folds*.

The main result is that the ranking of the strategies depends strongly on the validation scheme. Under the by-*chunk* protocol, S1 is the best model, with an F1 of 95.89, followed by S2, S4 and S3. Under the by-patient protocol, the ordering changes substantially: S4 becomes the best model, S2 and S3 obtain similar intermediate results, and S1 becomes the worst model, with an F1 of only 16.83. Therefore, the model that appears strongest when highly related *chunks* can be split between training and validation is the weakest once the evaluation is performed on unseen patients.

The clearest example of this effect is S1. With the by-*chunk* split, S1 reaches a recall of 98.07 and an F1 of 95.89. However, its recall drops to 14.10 under the by-*recording* split and to 12.12 under the by-patient split, while its specificity remains close to 99 in all three protocols. This pattern suggests that S1 benefits mainly from the near-duplicate ictal *chunks* created by the stride-1 sampling of the positive class. Once this temporal overlap is removed, its ability to detect ictal *windows* does not transfer.

The by-*recording* and by-patient results also show that avoiding overlap within the same recording is not sufficient to estimate clinical generalization. The by-*recording* split still allows the same subject to appear in both training and validation through different EDF files, whereas the by-patient split evaluates unseen subjects. For this reason, the remaining analysis focuses mainly on the by-patient protocol, which is the most demanding and the most relevant for patient-independent seizure detection.

B. Effect of the feature extractor

The by-patient results allow the contribution of each architectural choice to be compared under the strictest validation setting. Adding a per-channel CNN improves performance in both comparisons where it is introduced. Without a Transformer mixer, S1$\rightarrow$S2 raises F1 from 16.83 to 46.62 and recall from 12.12 to 53.86. With a Transformer mixer, S3$\rightarrow$S4 raises F1 from 47.05 to 60.53 and recall from 57.40 to 78.90. This indicates that compressing each channel into learned local temporal descriptors improves generalization to unseen patients.

The inter-channel Transformer also improves performance at both compression levels. On the raw signal, S1$\rightarrow$S3 increases F1 from 16.83 to 47.05 and recall from 12.12 to 57.40. On the CNN descriptors, S2$\rightarrow$S4 increases F1 from 46.62 to 60.53 and recall from 53.86 to 78.90. These results suggest that the Transformer mixer adds information beyond simple positional concatenation of channels.

The best patient-independent result is obtained by S4, which combines the two components. S4 reaches the highest recall (78.90) and the highest F1 (60.53) under the by-patient protocol. Its behaviour also contrasts with S1: while S1 collapses from F1 95.89 in the by-*chunk* split to 16.83 in the by-patient split, S4 degrades more gradually, from F1 85.62 to 60.53. This makes S4 the strongest feature extractor among the four evaluated strategies.

This improvement comes with a clear sensitivity–precision trade-off. S4 has the highest recall under the by-patient protocol, but also the lowest specificity (88.79) and only moderate precision (57.11). Therefore, S4 should be interpreted as the best generalizing model among those evaluated, not as a clinically deployable detector on its own. Its results show that the CNN + Transformer extractor is promising, but also that additional work is needed to reduce false positives.

C. Best and worst patients

The by-patient results show a strong variability across subjects. Using S4 (CNN + Transformer) as the reference model, the best-performing patients are chb19 ($R = 96.67,\ P = 88.55,\ F1 = 92.43$), chb03 ($R = 86.21,\ P = 93.58,\ F1 = 89.74$), chb11 ($R = 80.69,\ P = 97.90,\ F1 = 88.47$) and chb08 ($R = 77.06,\ P = 96.48,\ F1 = 85.68$). These subjects combine high recall and high precision, meaning that their ictal *windows* are detected while their interictal background remains well separated.

Table V: Validation metrics under **k-fold by *chunk*** ($K = 5$, 80/20 per *fold*). Each cell is the mean ± standard deviation across the five *folds* (*window* level, ictal vs. interictal). Most optimistic scheme (highest temporal leakage).

Strategy	Recall	Precision	Specificity	F1
S1 (raw + flatten)	**98.07±1.36**	**93.86±2.75**	**99.34±0.31**	**95.89±1.27**
S2 (CNN + flatten)	97.20±1.23	88.15±5.91	98.64±0.75	92.34±2.79
S3 (raw + Transformer)	96.40±2.44	70.67±12.60	95.57±2.66	80.91±7.81
S4 (CNN + Transformer)	89.40±2.67	82.86±9.68	97.98±1.52	85.62±4.61

Table VI: Validation metrics under **k-fold by *recording*** ($K = 5$). Each cell is the mean ± standard deviation across the five *folds* (*window* level, ictal vs. interictal). Intermediate strictness: *recordings* are split between training and validation without splitting any single recording, but the same subject may appear in both sets through different *recordings*.

Strategy	Recall	Precision	Specificity	F1
S1 (raw + flatten)	14.10±3.36	64.14±20.37	**99.20±0.43**	23.00±5.65
S2 (CNN + flatten)	80.88±7.65	**78.93±12.48**	97.66±1.85	**79.08±6.55**
S3 (raw + Transformer)	57.35±8.05	65.73±17.92	96.15±3.25	59.27±6.25
S4 (CNN + Transformer)	**89.18±6.26**	69.76±16.01	95.29±3.86	76.88±9.52

Table VII: Validation metrics under **group k-fold by patient** ($K = 5$, *leave-subjects-out*). Each cell is the mean ± standard deviation across the five *folds* (*window* level, ictal vs. interictal). Most demanding scheme (no leakage: no patient in both sets); the regime relevant to clinical generalization.

Strategy	Recall	Precision	Specificity	F1
S1 (raw + flatten)	12.12±8.74	37.25±13.75	**98.29±0.75**	16.83±10.18
S2 (CNN + flatten)	53.86±15.64	51.97±28.73	91.30±9.42	46.62±8.80
S3 (raw + Transformer)	57.40±17.77	46.80±17.31	92.40±4.84	47.05±5.70
S4 (CNN + Transformer)	**78.90±13.54**	**57.11±28.88**	88.79±15.18	**60.53±18.92**

At the opposite extreme, the worst cases follow two different patterns. Some patients are mainly missed, such as chb16 ($R = 2.78$, $P = 2.17$, $F1 = 2.43$) and chb06 ($R = 37.35$, $P = 9.87$, $F1 = 15.62$). In these cases, the model fails to detect a large fraction of the ictal *windows*. Other patients are over-detected, such as chb09 ($R = 100.00$, $P = 5.44$, $F1 = 10.33$) and chb10 ($R = 99.12$, $P = 14.99$, $F1 = 26.04$). Here the model detects the ictal *windows*, but produces too many false positives.

This suggests that the aggregate performance of S4 hides two distinct failure modes: low-recall patients, which may require better patient-independent representations, and high-recall/low-precision patients, which may require calibration or threshold adjustment.

D. Comparison with the across-subject reference study

The hierarchical validation adopted in this work is directly inspired by the across-subject study of Gil and Hernández-Sabaté [1], which evaluates six multi-channel *fusion* architectures on CHB-MIT at three levels of increasing strictness (window, state and subject). A direct numerical comparison is **not** meaningful, because the two works differ in the model families, in the preprocessing (1 s *windows* at 128 Hz with a 30 s peri-ictal exclusion in [1], versus 2 s *windows* at 256 Hz with 5 min *buffers* here) and in the class ratio kept at validation. What *can* be contrasted, and is the relevant comparison given that both studies share a methodological aim, is the direction and the rough magnitude of the degradation across protocols, summarized in Table VIII.

a) The over-optimism of sample-level validation is reproduced: The central finding of [1] is that window-level metrics are inflated by the lack of independence between training and test and collapse once the partition becomes subject-independent: their best fusion model drops in sensitivity from 96% at the window level to 77% at the subject level, and their recurrent baseline (M6, an LSTM) from 97% to 60%. Our experiments reproduce this effect under the equivalent pair of protocols (by *chunk* → by patient) and, for the weakest extractor, in an even more extreme form: the raw *baseline* S1 falls from 98.1% to 12.1% recall, and its F1 from 95.9 to 16.8. The agreement is therefore not only qualitative but also in the sign and severity of the collapse, which supports the validity of our pipeline as an independent reproduction of the phenomenon they describe.

b) The gap is governed by the channel-combination mechanism: A second point of convergence concerns *which* design choice drives across-subject performance. The study in [1] concludes that *fusion* mechanisms that preserve channel-specific information (concatenation, M1/M2, and the multiple weighted average, M5) generalize significantly better than shared-weight averages (M3/M4), i.e. that how the channels are combined is the decisive factor. Our design, which fixes

the downstream classifier and varies only the *feature* extractor, reaches a compatible conclusion from a different angle: the naive positional *flatten* (S1) collapses across subjects, whereas adding a per-channel CNN and, above all, an attention-based inter-channel mixer (S4) recovers most of the sensitivity, reaching 78.9% at the subject level—essentially the same range as their best fusion model M1 (77%) and above their LSTM M6 (60%). Both studies thus identify the multi-channel representation and its fusion as the main lever for across-subject generalization, above raw model capacity.

c) Per-patient heterogeneity and hard subpopulations: Both works report a strong inter-subject variability and isolate specific difficult cases. Through its GMM random-effect analysis, [1] identifies a hard subpopulation of eight subjects (*chb08*, *chb12–chb16*, *chb20–chb21*; 29% of the cases) whose ictal patterns degrade generalist models, with *chb14* as an extreme ictal outlier and *chb09* flagged for anomalous interictal (Recall−) behaviour. Our per-patient analysis (Table XIb, S4) independently surfaces several of the same subjects: *chb16* and *chb06* appear as low-recall (*missed*) cases and *chb09* and *chb10* as low-precision (*over-detected*) cases. The *chb09* agreement is especially direct: [1] marks it as an interictal outlier, which matches our finding of $R = 100\%$ but $P = 5.4\%$, i.e. a massive interictal false-positive rate. The main divergence is *chb08*, which is among the best-handled subjects in our setting ($P = 96.5\%$, F1 $= 85.7$) but is flagged as an interictal outlier in [1]; this is plausibly explained by the different preprocessing and by our per-channel CNN + attention extractor.

d) Increasing variance with strictness: Finally, both studies observe the variability of the metrics growing as the protocol becomes stricter. [1] quantifies the sensitivity standard deviation rising from 0.00 at the window level to 0.24 at the state level and 0.32 at the subject level. Our results show the analogous growth: for S4, the F1 standard deviation increases from 4.6 (by-*chunk fold*) to 18.9 (by-patient *fold*), with a per-patient F1 standard deviation of 26.5 at the subject level. In both studies, the strictest protocol is the one where mean performance is lowest and least stable, which is precisely the regime relevant for clinical deployment.

This convergence, obtained with different architectures and preprocessing, reinforces the methodological claim shared with [1]: reported seizure-detection performance is dominated by the validation protocol, and only subject-independent evaluation provides a realistic estimate of clinical generalization.

E. Main findings

Overall, the experiments support three main findings. First, the validation scheme has a major impact on the apparent performance of the models. Under the permissive by-*chunk* split, the raw baseline S1 obtains the best F1 score (95.89), but under the patient-independent split it becomes the worst strategy (16.83). This shows that chunk-level validation strongly overestimates generalization when highly overlapping ictal *chunks* can appear in both training and validation.

Second, under the strictest validation protocol, both architectural choices studied in this work contribute to patient-independent performance. Adding a per-channel CNN improves the representation of each EEG channel, while adding a Transformer mixer improves the aggregation of information across channels. Their combination in S4 gives the best result among the evaluated strategies, reaching the highest recall (78.90) and F1 score (60.53) under the by-patient split. This indicates that local temporal compression and attention-based inter-channel mixing are complementary.

Third, the remaining errors are highly patient-dependent. Some subjects are detected reliably, with high recall and precision, whereas others suffer from one of two different failure modes: missed ictal *windows*, where the model fails to recognize the seizure pattern, or excessive false positives, where the model over-fires on interictal activity. This heterogeneity shows that the aggregate metrics alone are not sufficient to understand the model's behaviour.

These findings define two natural directions for future work. The first is to evaluate richer feature extractors, especially the deferred self-supervised variants S5 and S6, to improve the representation of patient-specific seizure patterns. The second is to study patient-specific calibration or threshold adjustment, which may reduce false positives in subjects where recall is already high but precision remains low.

Table VIII: Degradation of sensitivity and specificity from the most permissive to the strictest protocol, contrasted with the across-subject reference study [1]. All values in %. **The two studies are not directly comparable** (different model families, preprocessing and validation class ratios); the table contrasts the *degradation trend*, not the absolute performance. Reference values read from Tables 2 and 6 of [1] (sensitivity = Recall+, specificity = Recall−).

Study / model	Metric	Permissive (win./chunk)	Subject-indep. (subj./patient)
[1], M1 (best fusion)	Sens.	96.0	77.0
	Spec.	100.0	90.0
[1], M6 (LSTM)	Sens.	97.0	60.0
	Spec.	100.0	91.0
This work, S1 (raw + flatten)	Sens.	98.1	12.1
	Spec.	99.3	98.3
This work, S4 (CNN + Transf.)	Sens.	89.4	78.9
	Spec.	98.0	88.8

IX. Follow-up on planning and methodological changes

With respect to the planning and methodology described in the initial report, the project has undergone a substantial reformulation of its architectural strategy.

The initial report proposed a single hierarchical *pipeline* consisting of a generic "spatial *encoder*" to fuse the channel information and a Transformer-based "temporal model", without specifying the concrete inter-channel fusion mechanism nor the representation-learning method. A first iteration of this plan was implemented as a fixed three-stage *pipeline* (per-channel CNN → spatial Transformer with CLS → temporal

Transformer), with the first two stages pretrained as reconstruction autoencoders.

During development it became clear that committing prematurely to a single instance of each stage left the contribution of each component unmeasured. For this reason, the project was reorganized around a comparative study: a fixed downstream classifier (LSTM + head) that operates over sequences of *window*-level *embeddings*, and six interchangeable *feature* extractors $g(\cdot)$ that vary systematically along two orthogonal axes (per-channel compression and inter-channel mixing). This reorganization preserves the original intuitions of the initial report (the value of a learned per-channel *encoder*, the value of inter-channel attention, the value of self-supervised pretraining), but turns each of them into a measurable hypothesis rather than a design assumption.

From an operational point of view, this reformulation also entailed:

- Reformulating the target task as **ictal vs. interictal** classification, excluding the preictal and postictal transition windows from the classification targets so that the two classes are cleanly separated.
- Replacing the original temporal Transformer with a 2-layer LSTM as the shared decision module, which is easier to train and stable under variable input dimensions.
- Shortening the decision *chunk* from 5 min (150 *windows*) to 60 s (30 *windows*; ictal *chunks* extracted with *stride* 1 and interictal *chunks* without overlap) to keep the LSTM in a regime in which it can be trained reliably with the available compute and data budget.
- Adopting **k-fold** cross-validation ($K = 5$) instead of a single *split*, and evaluating under three partition schemes of increasing strictness (by *chunk*, by *recording* and by patient), so that the metrics are reported as mean $\pm$ standard deviation across *folds* and the information leakage between sets can be quantified.
- Removing the pretraining of the spatial autoencoder: when a Transformer is used as the inter-channel mixer (S3, S4, S6), it is trained *end-to-end* as part of the classification task, rather than independently via reconstruction.

The original per-channel autoencoder is kept as a pretraining component of S5 and S6, now integrated within the comparative study rather than constituting the central axis of the project; its evaluation is deferred to a later iteration.

The dataset choice does not change: the project focuses on CHB-MIT, evaluated under the three k-fold partition schemes described (with the *leave-subjects-out* by patient being the most demanding); the use of additional datasets is deferred to later stages.

Acknowledgment

To my father, for passing on his curiosity about the world in the same way he taught me the meaning of perseverance, patience, and hard work, and for being a reference and an example in everything I do.

To my supervisors Aura and Debora, for introducing me to the world of research, and for always finding time to guide me when I needed it most. Beyond my supervisors, I consider them a model of what a researcher should be and aspire to.

References

[1] D. Gil and A. Hernández-Sabaté, "What is the across-subject performance of your Artificial Intelligence method for EEG analysis?," *Biomedical Signal Processing and Control*, vol. 120, p. 110020, 2026, doi: 10.1016/j.bspc.2026.110020.

[2] World Health Organization, "Epilepsy," Fact sheet, Feb. 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/epilepsy

[3] A. H. Shoeb, "Application of machine learning to epileptic seizure onset detection and treatment," Ph.D. dissertation, MIT, Cambridge, MA, 2009.

[4] T. Maiwald, M. Winterhalder, R. Aschenbrenner-Scheibe, H. U. Voss, A. Schulze-Bonhage, and J. Timmer, "Comparison of three nonlinear seizure prediction methods by means of the seizure prediction characteristic," *Physica D*, vol. 194, no. 3–4, pp. 357–368, 2004.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. ICLR*, 2021.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NeurIPS*, 2017, pp. 5998–6008.

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] H. Daoud and M. A. Bayoumi, "Efficient epileptic seizure prediction based on deep learning," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 5, pp. 804–813, Oct. 2019.

[9] N. D. Truong, A. D. Nguyen, L. Kuhlmann, M. R. Bonyadi, J. Yang, S. Ippolito, and O. Kavehei, "Convolutional neural networks for seizure prediction using intracranial and scalp electroencephalogram," *Neural Networks*, vol. 105, pp. 104–111, 2018.

[10] H. Khan, L. Marcuse, M. Fields, K. Swann, and B. Yener, "Focal onset seizure prediction using convolutional networks," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 9, pp. 2109–2118, Sep. 2018.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[12] A. L. Goldberger et al., "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, 2000.

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[14] T. Zhao, Y. Cui, T. Ji, J. Luo, W. Li, J. Jiang, Z. Gao, W. Hu, Y. Yan, Y. Jiang, and B. Hong, "VAEEG: Variational auto-encoder for extracting EEG representation," *NeuroImage*, vol. 304, p. 120946, 2024.

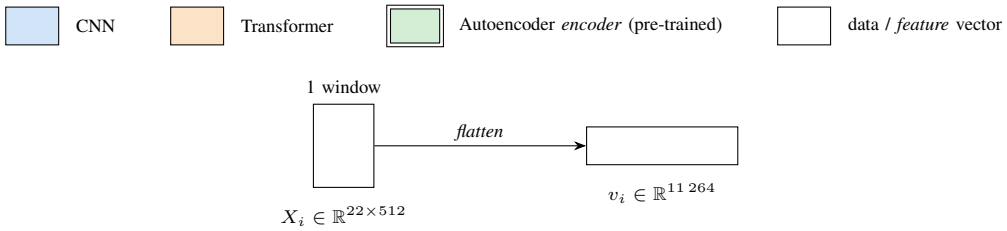
[15] J. Guttag, "CHB-MIT Scalp EEG Database (version 1.0.0)," PhysioNet, 2010. doi: 10.13026/C2K01R.

Appendix A
Diagrams of the six feature extractors

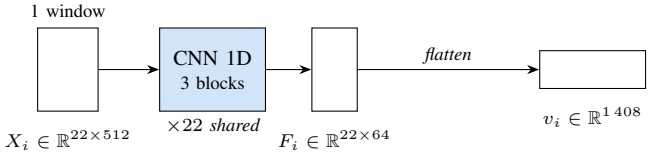
The shared prediction *pipeline* (Fig. 1) is the same for the six strategies; they differ only in the *feature* extractor $g(\cdot)$. Fig. 2 shows the six extractors side by side, each mapping a single *window* $X_i \in \mathbb{R}^{22\times512}$ to the *feature* vector v_i that is then fed to that shared *pipeline*.

Appendix B
Per-patient validation metrics

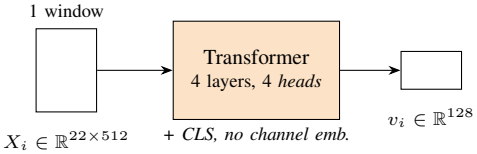
This appendix collects the per-patient validation metrics referenced in Section VIII, for the three partition schemes, ordered from least to most strict: by *chunk* (Table IX), by *recording* (Table X) and *leave-subjects-out* by patient (Table XI). Each split is shown on a single page, split into two blocks of two strategies (S1–S2 and S3–S4). Each row is computed on that subject's validation *chunks* (last epoch), as described in Section VII-C; all metrics in %; the last row of each block is the mean $\pm$ standard deviation across the 24 patients (values undefined for a given patient are excluded).



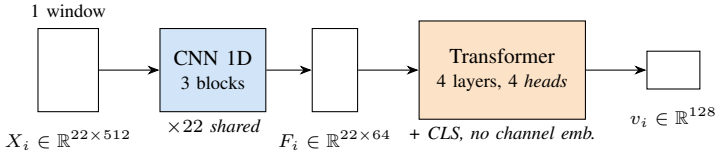
(a) S1: raw-signal *baseline* with *flatten*.



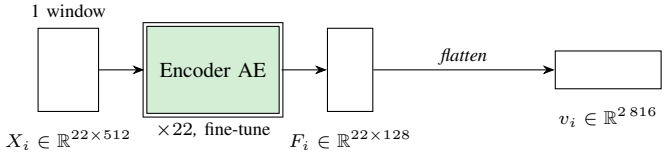
(b) S2: per-channel shared-weight CNN + *flatten*.



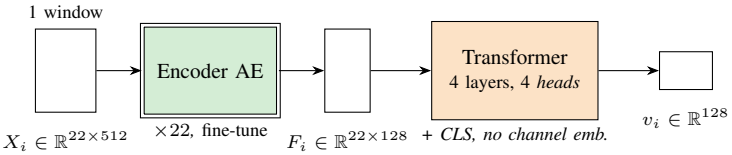
(c) S3: inter-channel Transformer on the raw signal.



(d) S4: per-channel CNN + inter-channel Transformer.



(e) S5: pre-trained autoencoder *encoder* + *flatten* (with *fine-tune*).



(f) S6: pre-trained autoencoder *encoder* + inter-channel Transformer (with *fine-tune*).

Figure 2: The six *feature* extractors $g(\cdot)$ compared in this work (S1–S6). Each maps a single *window* $X_i \in \mathbb{R}^{22\times512}$ to a *feature* vector v_i; only this extraction stage differs between the strategies. The downstream *pipeline* (*chunk* of 30 vectors $\rightarrow$ LSTM $\rightarrow$ head $\rightarrow$ p_i) is shared and identical for all six (Fig.~1). Learned modules are color-coded by family (CNN, Transformer, autoencoder *encoder*; see legend), while data tensors and *feature* vectors are shown in white. Identical color and shape across panels denote the same module reused in different case studies: the Transformer *encoder* (S3, S4, S6) and the per-channel CNN (S2, S4) are the same in every panel where they appear. *flatten* denotes vectorization (shown over the arrow), not a parametric module. A double border marks modules pre-trained in a self-supervised manner (S5, S6).

Table IX: Per-patient validation metrics under the **k-fold by *chunk*** scheme ($K = 5$), last epoch. Each subject appears in validation in *all* five *folds* (roughly 20 % of its *chunks* per *fold*); the per-patient confusion matrix accumulates those held-out *chunks* across the five *folds*, so each *chunk* is validated exactly once.

(a) S1 (raw + flatten) and S2 (CNN + flatten).

	S1 (raw + flatten)				S2 (CNN + flatten)			
Patient	**Recall**	**Precision**	**Specificity**	**F1**	**Recall**	**Precision**	**Specificity**	**F1**
chb01	97.31	93.13	99.32	95.18	97.31	94.76	99.49	96.02
chb02	93.10	96.43	99.86	94.74	91.95	94.12	99.76	93.02
chb03	96.06	97.01	99.73	96.53	98.03	94.76	99.50	96.37
chb04	99.48	76.31	99.37	86.36	93.72	76.82	99.42	84.43
chb05	98.58	86.88	98.16	92.36	98.58	92.05	98.95	95.21
chb06	97.59	81.82	99.54	89.01	87.95	39.89	97.18	54.89
chb07	98.78	95.29	99.80	97.01	98.78	79.41	98.95	88.04
chb08	98.92	99.13	99.65	99.02	98.92	95.21	97.97	97.03
chb09	100.00	81.03	99.18	89.52	96.45	80.95	99.20	88.03
chb10	99.12	93.39	99.45	96.17	97.81	90.65	99.21	94.09
chb11	99.26	98.28	99.66	98.77	99.50	99.26	99.85	99.38
chb12	99.21	95.10	97.51	97.12	98.62	82.98	90.13	90.13
chb13	97.32	93.16	97.11	95.20	95.09	84.52	92.96	89.50
chb14	91.11	91.11	99.46	91.11	84.44	75.25	98.31	79.58
chb15	98.81	97.83	98.96	98.32	98.71	96.31	98.21	97.49
chb16	77.78	87.50	99.58	82.35	80.56	41.43	95.65	54.72
chb17	100.00	93.63	99.14	96.71	96.60	76.76	96.31	85.54
chb18	99.39	89.01	99.01	93.91	93.87	86.44	98.81	90.00
chb19	100.00	98.36	99.88	99.17	98.33	93.65	99.53	95.93
chb20	94.04	91.03	99.11	92.51	98.68	88.17	98.73	93.13
chb21	91.18	93.94	99.69	92.54	99.02	97.12	99.84	98.06
chb22	99.05	94.55	99.67	96.74	100.00	98.13	99.89	99.06
chb23	99.07	96.40	99.47	97.72	98.61	94.25	99.14	96.38
chb24	96.05	96.81	99.30	96.43	92.09	81.47	95.34	86.46
Mean ± std	96.72±4.83	92.38±5.95	99.23±0.70	94.35±4.24	95.57±5.01	84.77±15.48	98.01±2.41	89.27±11.80

(b) S3 (raw + Transformer) and S4 (CNN + Transformer).

	S3 (raw + Transformer)				S4 (CNN + Transformer)			
Patient	**Recall**	**Precision**	**Specificity**	**F1**	**Recall**	**Precision**	**Specificity**	**F1**
chb01	96.86	83.08	98.13	89.44	95.96	93.04	99.32	94.48
chb02	94.25	58.99	97.27	72.57	91.95	80.81	99.09	86.02
chb03	96.55	73.41	96.78	83.40	86.70	90.26	99.14	88.44
chb04	96.86	36.42	96.53	52.93	88.48	58.28	98.70	70.27
chb05	98.58	75.75	96.10	85.67	97.16	75.27	96.05	84.83
chb06	85.54	25.91	94.80	39.78	49.40	31.54	97.72	38.50
chb07	98.78	46.42	95.31	63.16	97.56	66.67	97.99	79.21
chb08	98.92	92.70	96.83	95.71	92.42	96.39	98.59	94.36
chb09	98.58	42.12	95.25	59.02	93.62	70.21	98.61	80.24
chb10	98.25	58.79	94.63	73.56	98.25	82.35	98.36	89.60
chb11	97.77	83.33	96.13	89.98	88.86	95.73	99.22	92.17
chb12	90.96	92.05	96.17	91.50	90.96	79.01	88.22	84.57
chb13	93.75	90.52	96.03	92.11	75.00	91.80	97.29	82.56
chb14	87.78	39.50	91.80	54.48	72.22	59.63	97.02	65.33
chb15	99.30	81.67	89.42	89.63	94.63	93.69	96.98	94.16
chb16	75.00	22.88	90.35	35.06	75.00	38.57	95.44	50.94
chb17	99.32	57.03	90.57	72.46	91.84	72.58	95.63	81.08
chb18	96.32	67.09	96.18	79.09	82.82	86.54	98.96	84.64
chb19	96.67	67.44	96.72	79.45	88.33	92.98	99.53	90.60
chb20	94.04	79.78	97.71	86.32	77.48	69.23	96.70	73.12
chb21	97.06	71.22	97.92	82.16	62.75	86.49	99.48	72.73
chb22	99.05	60.47	96.28	75.09	96.19	77.69	98.41	85.96
chb23	99.54	73.13	94.78	84.31	96.76	77.12	95.90	85.83
chb24	88.54	83.27	96.05	85.82	77.87	89.14	97.89	83.12
Mean ± std	94.93±5.78	65.12±20.58	95.32±2.40	75.53±16.67	85.93±12.29	77.29±17.09	97.51±2.35	80.53±13.51

Table X: Per-patient validation metrics under the **k-fold by *recording*** scheme ($K = 5$), last epoch. The *recordings* are split across the five *folds* without regard to the patient; each *recording* is held out in exactly one *fold*, so the per-patient confusion matrix accumulates the subject's held-out *recordings* across the *folds*.

(a) S1 (raw + flatten) and S2 (CNN + flatten).

	S1 (raw + flatten)				S2 (CNN + flatten)			
Patient	**Recall**	**Precision**	**Specificity**	**F1**	**Recall**	**Precision**	**Specificity**	**F1**
chb01	26.46	78.67	99.32	39.60	94.17	91.30	99.15	92.72
chb02	0	0	99.23	0	89.66	83.87	99.28	86.67
chb03	0	0	99.68	0	95.07	96.02	99.64	95.54
chb04	40.31	59.23	99.43	47.98	39.27	22.06	97.15	28.25
chb05	6.38	32.73	98.38	10.68	90.78	95.52	99.47	93.09
chb06	1.20	5.00	99.51	1.94	36.14	30.30	96.94	32.97
chb07	29.27	76.19	99.62	42.29	96.95	59.33	97.27	73.61
chb08	14.07	92.86	99.56	24.44	76.19	93.12	97.71	83.81
chb09	26.95	45.78	98.88	33.93	88.65	33.24	93.76	48.36
chb10	27.19	87.32	99.69	41.47	97.81	91.02	99.25	94.29
chb11	38.12	90.59	99.22	53.66	95.79	99.74	99.95	97.73
chb12	0.78	13.79	97.61	1.48	71.51	85.65	94.16	77.94
chb13	4.91	57.89	98.56	9.05	57.14	76.19	92.78	65.31
chb14	0	0	99.86	0	66.67	53.57	96.47	59.41
chb15	17.41	77.09	97.54	28.41	79.60	92.27	96.84	85.47
chb16	0	0	99.36	0	63.89	27.38	93.53	38.33
chb17	10.88	45.71	98.37	17.58	41.50	50.41	94.85	45.52
chb18	1.84	10.71	98.76	3.14	72.39	82.52	98.76	77.12
chb19	13.33	61.54	99.41	21.92	80.83	93.27	99.59	86.61
chb20	7.28	40.74	98.98	12.36	80.79	69.71	96.63	74.85
chb21	0	0	99.69	0	60.78	92.54	99.74	73.37
chb22	3.81	44.44	99.73	7.01	95.24	99.01	99.95	97.09
chb23	9.25	64.52	99.27	16.19	98.15	88.70	98.22	93.19
chb24	17.00	81.13	99.12	28.10	78.66	79.28	95.43	78.97
Mean ± std	12.35±12.77	44.41±33.40	99.12±0.63	18.38±17.39	76.98±19.17	74.42±25.06	97.36±2.26	74.18±21.32

(b) S3 (raw + Transformer) and S4 (CNN + Transformer).

	S3 (raw + Transformer)				S4 (CNN + Transformer)			
Patient	**Recall**	**Precision**	**Specificity**	**F1**	**Recall**	**Precision**	**Specificity**	**F1**
chb01	87.89	82.35	98.22	85.03	96.41	88.11	98.77	92.08
chb02	41.38	60.00	98.85	48.98	87.36	74.51	98.76	80.42
chb03	61.08	72.51	97.87	66.31	91.63	80.52	97.96	85.71
chb04	73.30	27.61	96.06	40.11	88.48	33.14	96.34	48.22
chb05	83.69	73.75	96.32	78.41	97.52	63.66	93.12	77.03
chb06	0	0	97.39	0	25.30	9.81	95.06	14.14
chb07	93.29	48.57	95.94	63.88	98.78	41.97	94.38	58.91
chb08	64.29	87.61	96.30	74.16	85.50	93.16	97.44	89.16
chb09	77.30	37.33	95.45	50.35	93.62	37.93	94.63	53.99
chb10	84.65	49.61	93.30	62.56	99.12	56.36	94.02	71.86
chb11	82.18	92.48	98.68	87.02	96.78	93.76	98.73	95.25
chb12	13.75	55.56	94.64	22.05	93.71	69.43	79.89	79.77
chb13	29.02	81.25	97.29	42.76	75.89	83.74	94.04	79.63
chb14	4.44	3.96	93.42	4.18	71.11	45.07	94.71	55.17
chb15	56.52	75.83	91.45	64.77	92.04	76.89	86.87	83.79
chb16	13.89	4.50	88.76	6.80	75.00	26.47	92.05	39.13
chb17	39.46	41.73	93.05	40.56	97.96	63.16	92.80	76.80
chb18	49.08	57.97	97.13	53.16	82.21	76.57	97.97	79.29
chb19	78.33	65.28	97.07	71.21	90.83	78.99	98.30	84.50
chb20	37.09	55.45	97.14	44.44	83.44	48.09	91.36	61.02
chb21	32.35	56.90	98.70	41.25	80.39	77.36	98.75	78.85
chb22	75.24	58.09	96.88	65.56	100.00	63.64	96.72	77.78
chb23	50.93	70.06	96.90	58.98	97.22	69.77	93.99	81.24
chb24	73.91	78.24	95.43	76.02	78.66	77.43	94.90	78.04
Mean ± std	54.29±28.14	55.69±25.87	95.93±2.45	52.02±24.48	86.62±15.61	63.73±22.06	94.65±4.28	71.74±18.77

Table XI: Per-patient validation metrics under the **group k-fold by patient** (*leave-subjects-out*) scheme ($K = 5$), last epoch. Each subject is held out for validation in exactly one *fold*; values are computed on that subject's validation *chunks* (ictal vs. interictal).

(a) S1 (raw + flatten) and S2 (CNN + flatten).

	S1 (raw + flatten)				S2 (CNN + flatten)			
Patient	**Recall**	**Precision**	**Specificity**	**F1**	**Recall**	**Precision**	**Specificity**	**F1**
chb01	23.32	78.79	99.41	35.99	92.83	83.81	98.30	88.09
chb02	17.24	55.56	99.43	26.32	90.80	86.81	99.43	88.76
chb03	0	0	99.09	0	44.33	90.00	99.55	59.41
chb04	42.41	28.42	97.81	34.03	61.26	28.54	96.85	38.94
chb05	12.41	22.73	94.78	16.06	89.36	44.21	86.06	59.15
chb06	8.43	3.72	95.36	5.16	42.17	9.23	91.19	15.15
chb07	4.26	19.44	99.27	7.00	98.78	37.67	93.28	54.55
chb08	1.73	47.06	99.21	3.34	52.81	96.06	99.12	68.16
chb09	43.97	43.66	98.01	43.82	95.74	13.43	78.38	23.56
chb10	32.89	43.35	96.65	37.41	56.14	7.50	46.07	13.24
chb11	18.56	84.27	99.31	30.43	77.97	86.30	97.55	81.92
chb12	0.58	37.5	99.52	1.16	9.03	37.10	92.53	14.53
chb13	5.35	37.5	96.39	9.37	6.25	82.35	99.46	11.62
chb14	0	0	99.80	0	0	0	99.73	0
chb15	10.55	81.54	98.87	18.68	42.39	30.04	53.16	35.16
chb16	0	0	99.36	0	8.33	5.45	94.49	6.59
chb17	19.05	41.79	96.66	26.17	46.94	41.57	91.68	44.09
chb18	4.29	17.07	98.32	6.86	71.78	80.69	98.61	75.97
chb19	0	0	99.59	0	65.00	88.64	99.41	75.00
chb20	7.94	22.64	97.40	11.76	15.89	28.57	96.19	20.43
chb21	1.96	5.55	98.23	2.89	55.88	60.64	98.08	58.16
chb22	0.95	5.88	99.71	1.63	56.19	96.72	99.89	71.08
chb23	6.94	38.46	98.41	11.76	85.19	65.48	93.59	74.04
chb24	9.88	96.15	99.91	17.92	69.96	84.69	97.19	76.62
Mean ± std	11.36±12.99	33.79±29.12	98.35±1.45	14.49±13.95	55.63±30.42	53.56±33.29	91.66±13.93	48.09±28.84

(b) S3 (raw + Transformer) and S4 (CNN + Transformer).

	S3 (raw + Transformer)				S4 (CNN + Transformer)			
Patient	**Recall**	**Precision**	**Specificity**	**F1**	**Recall**	**Precision**	**Specificity**	**F1**
chb01	81.17	72.69	97.11	76.69	97.31	64.20	94.86	77.36
chb02	89.66	44.57	95.35	59.54	89.66	72.90	98.61	80.41
chb03	43.35	52.07	96.33	47.31	86.21	93.58	99.46	89.74
chb04	79.06	13.58	89.68	23.18	90.58	31.45	95.95	46.69
chb05	92.55	67.44	94.48	78.03	99.65	25.87	64.71	41.08
chb06	21.69	7.28	94.14	10.91	37.35	9.87	92.75	15.62
chb07	84.15	40.35	94.88	54.55	98.78	39.90	93.88	56.84
chb08	41.56	75.00	94.36	53.48	77.06	96.48	98.85	85.68
chb09	83.69	10.64	75.37	18.88	100.00	5.44	39.14	10.33
chb10	75.88	18.72	74.32	30.03	99.12	14.99	56.16	26.04
chb11	82.92	89.81	98.14	86.23	80.69	97.90	99.66	88.47
chb12	19.84	73.72	96.55	31.27	41.45	84.74	96.36	55.67
chb13	15.18	73.91	97.83	25.19	42.41	88.79	97.83	57.40
chb14	0	0	95.73	0	41.11	42.53	96.61	41.81
chb15	37.91	78.72	95.14	51.18	87.06	56.49	68.18	68.52
chb16	0	0	91.73	0	2.78	2.17	95.23	2.43
chb17	83.67	47.49	88.34	60.59	99.32	62.39	92.45	76.64
chb18	63.19	55.38	95.89	59.03	81.60	85.81	98.91	83.65
chb19	60.00	47.37	95.31	52.94	96.67	88.55	99.12	92.43
chb20	58.28	39.11	91.30	46.81	65.56	52.11	94.22	58.06
chb21	48.04	40.50	96.26	43.95	30.39	64.58	99.12	41.33
chb22	85.71	56.25	96.17	67.92	97.14	70.34	97.65	81.60
chb23	66.67	57.37	92.93	61.67	100.00	64.67	92.21	78.55
chb24	70.75	80.27	96.13	75.21	79.84	66.45	91.04	72.53
Mean ± std	57.71±29.10	47.59±27.12	93.06±6.11	46.44±24.28	75.91±27.91	57.59±29.99	89.71±15.84	59.54±26.50