SUPPLEMENTAL MATERIAL

Development of Expert-level Classification of Seizures and Other Highly Epileptiform Brain Activity During EEG Interpretation

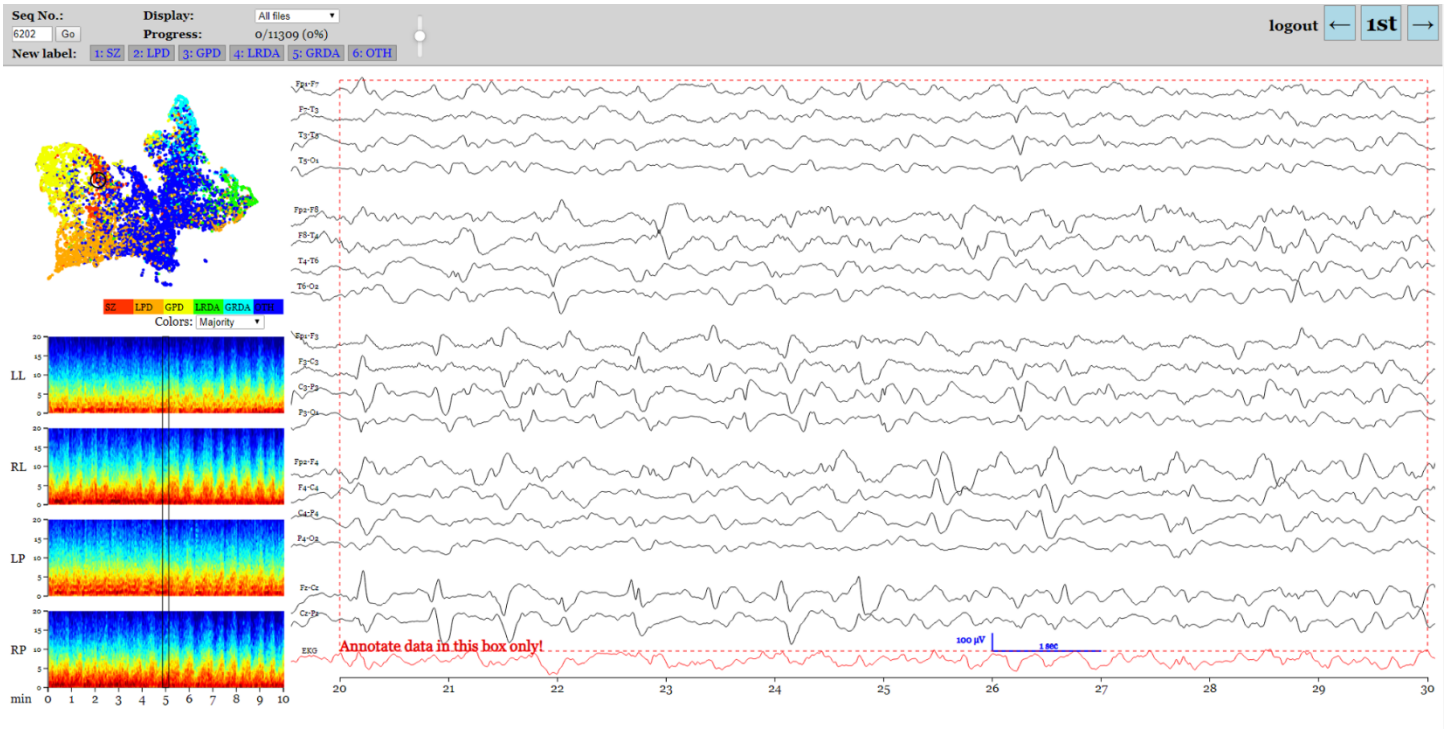
This document includes additional details about the following.

Number of Supplemental Tables: 1
Number of Supplemental Figures: 9

S1. Data labeling

The approach used to collect labels from multiple experts is described in the companion paper. A screenshot of the web-based graphical user interface (GUI) to collect annotations from experts is shown in **Figure S1.**

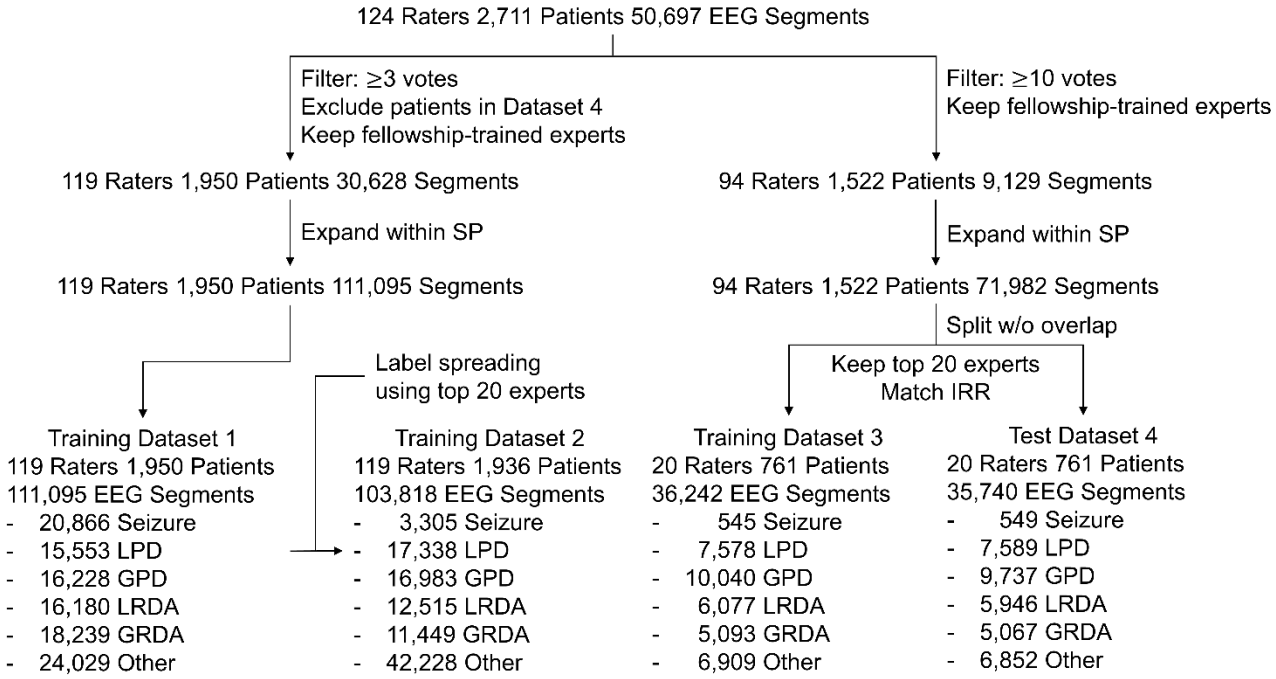
Figure S1. Web-based GUI used to collect annotations of EEG segments from multiple experts.



S2. Creation of training and test datasets

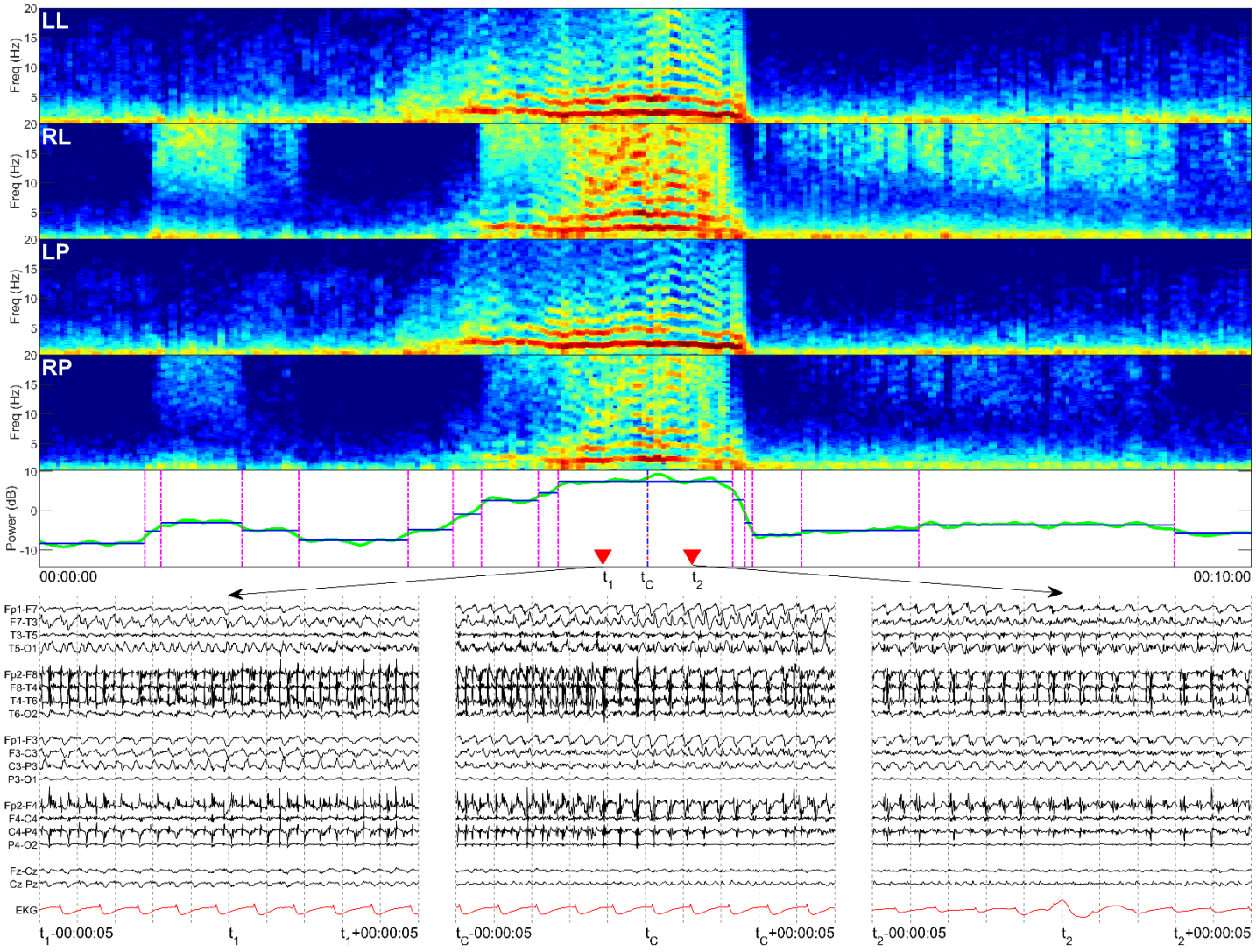
A flow diagram for creation of training and test datasets is shown in **Figure S2**. See also Table 1 in the main text.

Figure S2. Flow diagram for creation of training and test datasets.



Overall, 124 raters scored 50,697 EEG segments from 2,711 distinct patients' EEGs. Among the 124 raters, we identified a subset of 20 physician experts with subspecialty training who each individually annotated ≥1000 EEG segments. Among EEG segments annotated by these 20 experts, 9,129 segments received labels from at least 10 independent experts. Based on prior work suggesting that ~10 experts are required to achieve a stable group consensus[1], we designated these segments as having "high quality" labels, meaning that these samples received enough expert labels so that the consensus label (IIIC pattern type with the most labels) and degree of agreement amongst experts about the correct label can be assigned with high confidence. The remaining labeled data was considered "lower quality" e.g., because these segments either had fewer labels or not all raters had fellowship training. We therefore divided the 50,697 labeled EEG segments into a group of 9,129 segments with "high quality" labels scored by 20 "top experts"; the remaining set of 30,628 segments had lower-quality labels collectively received from 119 of the 124 raters.

Figure S3: Samples belonging to the same stationary period (SP) are assigned the same label.



We expanded the high- and low-quality sets of EEG segments by adding additional segments belonging to the same stationary period of the EEG. This yielded 71,982 segments with high-quality labels from 1,522 patients, and 111,095 segments with lower-quality labels from 1,950 patients. Nevertheless, in the final division of data into training and test datasets, any given patient's data is assigned exclusively to either the training or to the test dataset.

The justification for expanding labeled data is as follows: Empirically, the EEG can be divided into a series of "stationary periods" (SP), within which the pattern of EEG activity is unchanging. These SP can be identified by detecting changepoints within the EEG power, as illustrated in **Figure S3.** The upper four subplots of **Figure S3**

show spectrograms from four brain regions (LL = left lateral, RL = right lateral, LP = left para-central, and RP = right para-central). A seizure occurs near the center of the image. Below the spectrogram is shown the sum of the total spectral power across all four regions, and the SP between changepoints (CPs, identified by a CP detection algorithm). Raw EEG from three different segments within the SP (region within the central part of the seizure, between the two central pink vertical lines) are shown below, indicated by their starting times (t_1, t_C, t_2), where "t_C" represents the time at the center of SP shown. Each 10-sec EEG segment within the same SP demonstrates clear seizure activity, like the central segment. This example illustrates the rationale for assigning the same label to all EEG segments that fall within the same SP. In our approach, experts label the central EEG segment, and the same label is then automatically assigned to the remaining segments within the SP, increasing the number of labeled samples available for model training and evaluation.

After expanding the datasets, we further divided the dataset into "Datasets 1-4" as follows.

1- High-quality labels: The set of EEG segments with high quality labels was split into 2 sets called "Dataset 3" which was included as part of the training dataset, and "Dataset 4" which was included as part of the test dataset. These datasets were created to be balanced with respect to the number of patients, relative proportions of each type of IIIC pattern, and the level of agreement among experts about the correct labels. The process for achieving this balance is described below (see below "*Procedure for balancing the training and test datasets"*). Dataset 3 contained segments from 761 distinct patients: Dataset 4 contained EEG segments from 761 other distinct patients. No patients overlapped between Dataset 3 and 4.
2- Lower-quality labels: There were 111,095 EEG segments with lower-quality labels. Each had a minimum of 3 independent scores, collectively from 119 different raters. We designate this as "Dataset 1". We additionally created a subset of 103,818 of these EEG segments that we call "Dataset 2", with "pseudo-labels" in place of the labels directly obtained by the raters. The process for creating these pseudo-labels is described in a subsequent section (see below, "*Pseudo label generation*").

As described below (see "*Multi-step model training"*), Datasets 1-3 are used in different stages of training *SzNet.*

High quality labels: We used a stratified random sampling procedure to split the 71,982 EEG segments with high-quality annotations into approximately equal-sized training and test sets, such that the training and test data satisfied the following conditions, as shown in Table 1 of the main text:

- *All data from any given patient appeared entirely in the training or in the test set*. In other words, data was assigned to training and test sets at the patient level. This is important to avoid the possibility of over-estimating model performance in case the model over-learns patterns from any patient.
- *Training and test datasets have equal numbers of patients.*
- *Training and test datasets have approximately equal numbers and proportions of each IIIC class.*
- *Training and test datasets show approximately equal agreement among experts for each IIIC class.* This was to help ensure that the training and test datasets were comparable in terms of difficulty. This balancing requirement was achieved by ensuring approximately equal true positive rates (TPR, aka sensitivity), false positive rates (FPR), and positive predicted values (PPV, aka precision) for all IIIC classes.

Procedure for balancing the training and test datasets: Balancing of the training and test datasets was accomplished using the following iterative procedure:

1. Initially assign all data from each patient with high quality labels to either the training or test dataset at random. This ensures that training and test datasets include data from equal numbers of patients.
2. Calculate the following absolute differences in the number of each IIIC class, and in expert performance statistics. That is: Let $N_{train}(i)$ = number of segments with consensus label equal to pattern i, for $i = 1, 2, \ldots, 6$ (representing the 6 IIIC classes); and defined $N_{test}(i)$ similarly. Also define $TPR_{train}(i)$ = average rate of true positives among experts in classifying segments with consensus label i, and define $FPR_{train}(i)$, $PPV_{train}(i)$, $TPR_{test}(i)$, $FPR_{test}(i)$, and $PPV_{test}(i)$ similarly. In each case, the "ground truth" is taken to be the consensus label among all labels *except* those from the expert being evaluated. Then calculate the following quantities:

$$
\begin{aligned}
dN &= \max_i |N_{train}(i) - N_{test}(i)| \\
dTPR &= \max_i |TPR_{train}(i) - TPR_{test}(i)| \\
dFPR &= \max_i |FPR_{train}(i) - FPR_{test}(i)| \\
dPPV &= \max_i |PPV_{train}(i) - PPV_{test}(i)| \\
d &= dN + dTPR + dFPR + dPPV
\end{aligned}
$$

3. Randomly select one patient from the training dataset, and one from the test dataset to perform a proposed swap: Temporarily move all data from the first patient from the training to the test dataset and move all data from the second patient from the test dataset to the training dataset. Calculate new values resulting from this proposed swap: $dN', dTPR', dFPR', dPPV', d'$. If value $d' < d$, then accept the proposed swap, because it brings the training and test datasets closer to the desired balance.
4. Repeat steps 3 and 4 until d stops decreasing.

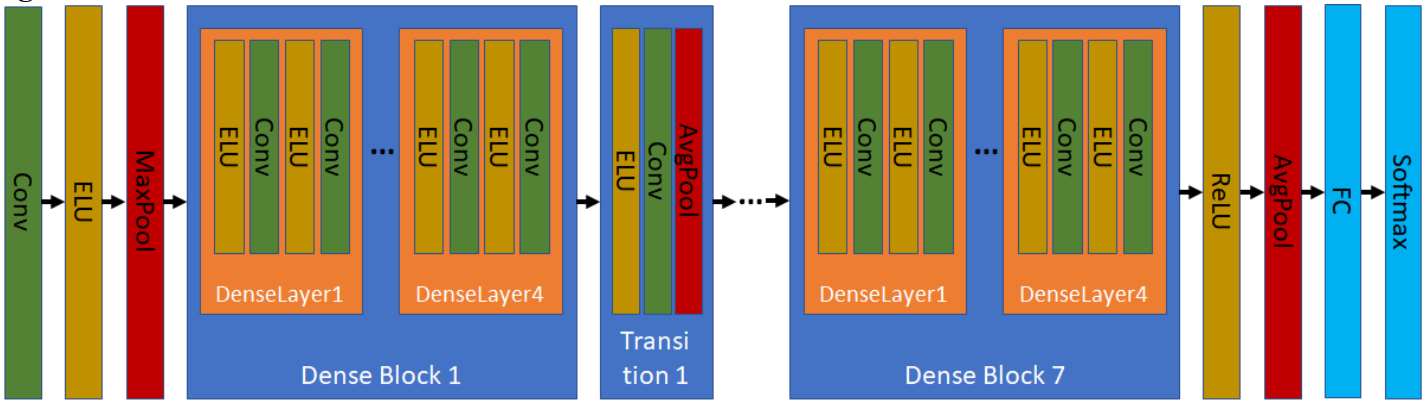
In our data, this procedure can produce excellent balance (to within ~1% on all component parts of d) after ~500 iterations. The split results are shown in Table 1 in the main text.

Data augmentation with lower-quality labels: We used the remaining lower-quality 111,095 EEG segments with labels from ≥3 experts, and a subset of 103,818 segments from top 20 experts, to further augment the training set. No data from these patients was included in the test dataset.

S3. Model architecture

A Dense-Net[2] type of Convolutional Neural Network (CNN)[3] was developed to classify IIIC patterns (**Figure S4**). The architecture of the Dense-Net used is as follows: There are 7 dense blocks. Each dense block includes 4 dense layers. Each dense layer includes 2 convolutional layers and 2 exponential linear unit (ELU) activations. Additionally, there are 6 transition blocks among the 7 dense blocks, where each transition block includes ELU activation, a convolutional layer, and an average pooling layer. The second-to-last layer is fully connected (FC) layer. The last layer is 6-dimensional SoftMax layer for the 5 IIIC patterns and the "Other" class.

Figure S4. Architecture of *SzNet,* based on the Dense-Net CNN architecture.



S4. Model training

In training *SzNet*, we use a combination of one-hot / "hard" labels, and soft-labels (see below, "*Multi-step model training")*.

Soft labels:

We denote the number of human experts as I, the number of labeled EEG segments as K, and the number of label categories as M. We denote the vote of the i-th human expert for the k-th EEG segment as $v_{i,k}$, where $v_{i,k} \in$

$\{NaN, 1,2, \cdots, M\}$, and NaN means that the i-th human expert did not cast a vote for the k-th EEG segment. Thus, the k-th EEG segment is given a label vector, containing the number of votes received for each pattern class across all experts, denoted $L_k = [l_{k,1}, l_{k,2}, \cdots, l_{k,M}]$, where

$$l_{k,m} = \sum_{i=1}^{I} \mathbb{I}(v_{i,k}, m), \quad \mathbb{I}(v_{i,k}, m) = \begin{cases} 1 & if\ v_{i,k} = m \\ 0 & otherwise \end{cases} \qquad (1)$$

We define the normalized label vector (soft label) for the k-th EEG segment as $\tilde{L}_k = [\tilde{l}_{k,1}, \tilde{l}_{k,2}, \cdots, \tilde{l}_{k,M}]$, where

$$\tilde{l}_{k,m} = \frac{l_{k,m}}{\sum_{i=1}^{M} l_{k,i}} \qquad (2)$$

Hard labels:

We define the one-hot vector (hard label) for the k-th EEG segment as $\acute{L}_k = [\acute{l}_{k,1}, \acute{l}_{k,2}, \cdots, \acute{l}_{k,M}]$, where

$$\acute{l}_{k,m} = \begin{cases} 1 & if\ m = \arg\max_{m} l_{k,m} \\ 0 & otherwise \end{cases}$$

EEG segments labeled by experts were represented by one hot vectors / hard labels, based on the majority vote among experts, with the rationale that this strategy is more likely to produce a model that accurately predicts the correct label (which is defined by expert consensus). By contrast, soft labels were used to represent label information given to unlabeled data (pseudo-labels), to represent the uncertainty inherent in the process of creating pseudo-labels via label propagation. The method for creating pseudo-labels via "label spreading" within the embedding map is described below (see "*Pseudo label generation"*).

Training objective function:

The goal in training *SzNet* is that the predicted labels should be close to the labels given by human experts. To formalize this goal, we use the Kullback-Leibler (KL) divergence as the cost, also known as the relative entropy[4,5]. KL divergence is used to measure the distance between the label distribution in the training data vs the trained *SzNet* model. We denote the predicted normalized score vector from the model as $\hat{L}_k = [\hat{l}_{k,1}, \hat{l}_{k,2}, \cdots, \hat{l}_{k,M}]$. The KL divergence between the normalized expert label vector $\tilde{L}_k$ and the normalized predicted label vector $\hat{L}_k$ is defined as

$$KL_{average} = \frac{1}{K}\sum_{k=1}^{K} KL(\tilde{L}_k, \hat{L}_k) = \frac{1}{K}\sum_{k=1}^{K}\sum_{m=1}^{M}\left[\tilde{l}_{k,m} \cdot \log\frac{\tilde{l}_{k,m}}{\hat{l}_{k,m}}\right].$$

Gradient descent settings: (Parameter and hyper-parameter settings)

Model parameter values for *SzNet* were optimized to minimize the KL divergence using gradient descent on the training data set. Adam (an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments) was used as the optimizer for the CNN.[6] The minibatch size for gradient descent was set to 256. Each EEG segment included 16 L-bipolar channels and 10-second EEG signal. The EEG sampling rate was 200Hz, thus the dimensions of the input tensor for the CNN were (256, 16, 2000). The dimensions of the representation vector in the second to last layer was set to be (255,1). The initial learning rate for gradient descent was 0.0001. The python libraries we used included *pytorch*, *sklearn*, *scipy*, *numpy*.

Regularization strategies

We employed the following strategies to reduce the chances of overfitting during model training:

- *Data augmentation:* We augmented the training data by "channel flipping", in which a duplicate of each EEG segment is created by exchanging the corresponding left- and right-sided EEG channels. The rationale for this strategy is as follows. Consider an EEG segment labeled LPD by most experts, where

the LPDs occur on the left side of the brain. If the same pattern occurred on the right side of the brain, it should still be called LPD, because the definition of LPDs includes patterns on either side. The same is true for the other 4 IIIC patterns, and for the "Other" class.

- *Weight decay (L2 regularization)*: We added a squared penalty term to the objective function to penalize large weights, i.e., we used L2 regularization. The weight decay parameter was set to 0.001.
- *Dropout*: During each training epoch some CNN nodes were randomly dropped out with a probability of 0.2.

Pseudo label generation

We augmented the training data by creating pseudo-labels (**Figure S5, A**) for a subset of the EEG segments in Dataset 1; this subset with its accompanying pseudo-labels is called Dataset 2 (**Figure S2**). The process for creating Dataset 2 is as follows. First, we created 20 embedding maps (UMAPs) for each of the top 20 experts (**Figure S5, B**). Within each expert's embedding map, we used "label spreading" from the EEG segments that the expert labeled to assign "pseudo-labels" to the segments that they had not labeled. For each unlabeled EEG segment, its pseudo-label was simply the label of the labeled segment that was nearest to it based on Euclidean distance within the UMAP. In this way, labels are spread from the labeled EEG segments to all the unlabeled segments, so that each of the EEG segments receives at least one label (pseudo- or real) from each expert.

For each EEG segment, we create a count vector by adding all labels assigned to it by the 20 experts (real labels and pseudo-labels) and any labels from the remaining 104 raters, to create a count vector (see equation (1) above). The count vector was then normalized (equation (2)) to produce a probability vector in which the fraction of votes assigned to each class is as an estimate of the probability of that class.

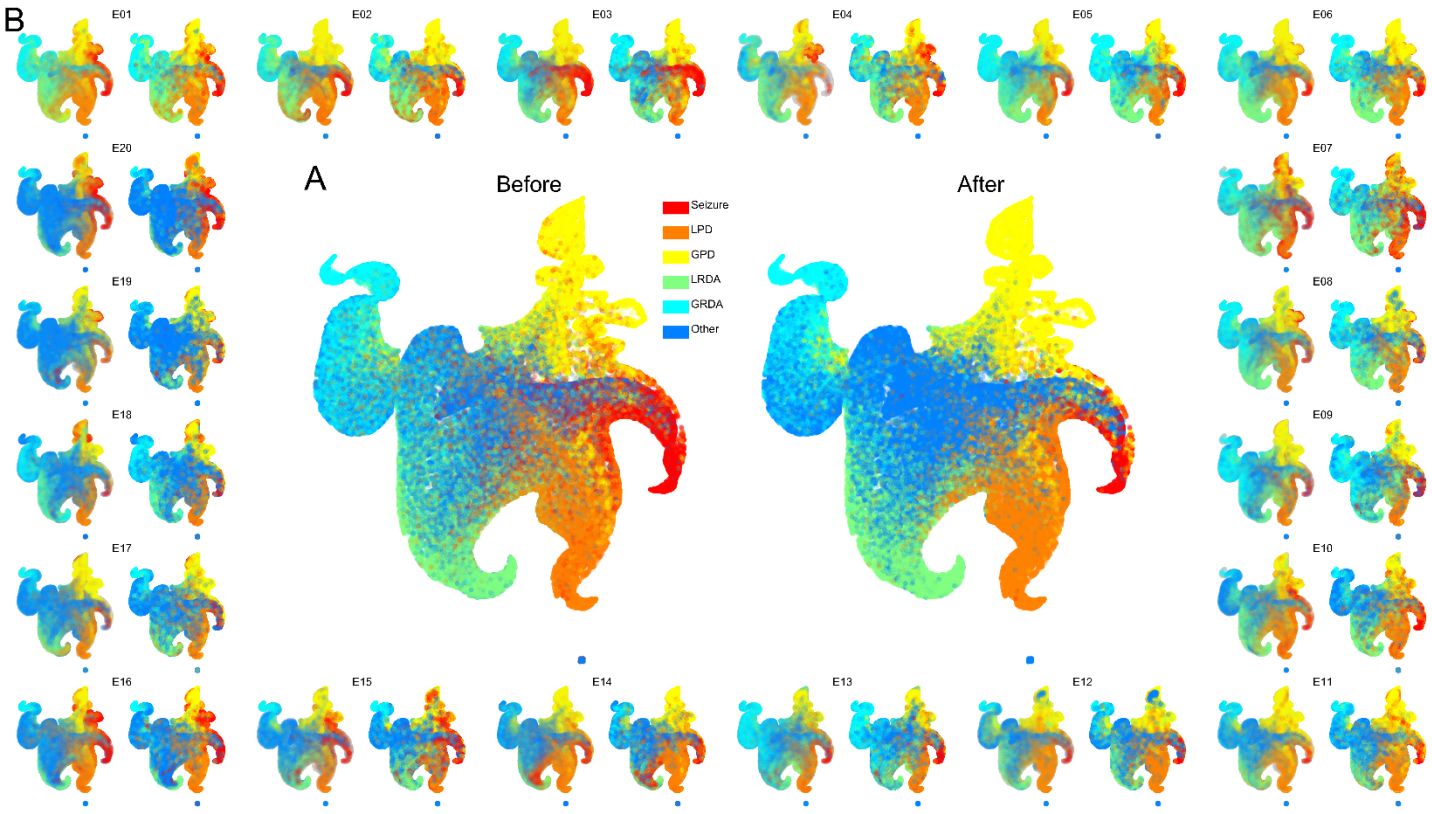


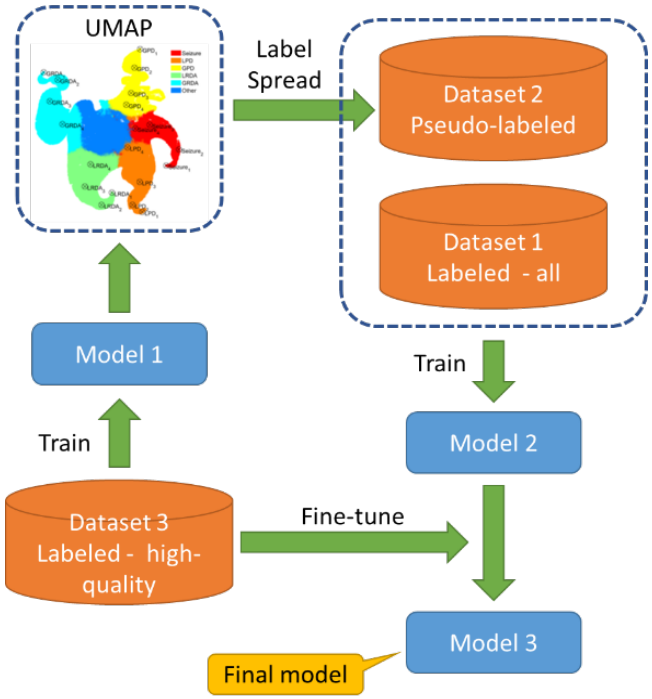
Figure S5. Creation of pseudo-labels via "label spreading" in Steps 3-4 of the model development procedure for *SzNet*. A: The embedding map ("UMAP") produced by the CNN trained in Step 1. "Real" labels from the 20 different experts are indicated by the colors; however, not all experts labeled all the points. To augment the training data, an individual UMAP was created for each expert and used to "spread" labels from the points that each expert did label (left hand UMAP in each of the pairs shown in B) to the points that the expert did not label

(right-hand UMAP of each pair). These UMAPs were then averaged to create the single overall UMAP illustrated on the right in A. Note that the averaging process produces "soft" labels, whereas for illustration purposes the final UMAP depicted above is colored based on the IIIC pattern whose label has the highest value after label spreading.

Multi-step model training
To train *SzNet*, we adopted a 6-step training strategy, as follows. This training strategy is illustrated in **Figure S6.**

- **Step 1**: We trained a first CNN using the high-quality labeled training EEG segments ("Dataset 3").
- **Step 2**: We used the first CNN (Model 1) to scan the high-quality labeled segments ("Dataset 3") and all remaining segments ("Dataset 1") to generate vector representations of these segments (activation values of the second to last layer in the CNN). We generated a single 2D embedding map based on these vector representations using the UMAP algorithm[7] (**Figure S5A,** "Before").
- **Step 3**: Using the UMAP, we generated pseudo-labels following the procedure described to create "Dataset 2" (**Figure S5B**). Note that Dataset 1 and Dataset 2 contain largely the same EEG segments, but with different labels: Dataset 1 has "direct labels" from experts, but generally fewer than 10 labels per segment – they are not "high quality" direct labels. By contrast, the pseudo-labels in Dataset 2 are an attempt to simulate obtaining labels for a very large number of samples from each of many experts. These two datasets are complementary, and both are used into training of Model 2.
- **Step 5**: We trained a second CNN (Model 2) using both Dataset 1 and Dataset 2.
- **Step 6**: We fine-tuned the second CNN by once again using the high-quality labels (Dataset 3). This final CNN, designated as Model 3 in **Figure S6**, is called *SzNet*.

Figure S6. Model development for *SzNet*.



S5. Discrimination
The ability of *SzNet* to discriminate between IIIC patterns was evaluated using Receiver Operating Characteristic (ROC) curves and with Precision Recall (PR) curves. Because experts classified EEG segments but did not provide class probabilities, rather than the ROC and PR curves we calculate an *operating point* for each expert on the ROC and PR plots and compare *SzNet* with the group of experts. Results are shown in the main text, **Figure 1A, B**. The methods for creating and interpreting these plots are described here.

ROC curves and expert operating points:
The ROC curve (**Figure 1A**) shows the relationship between *SzNet*'s true positive rate (TPR), aka sensitivity, and its false positive rate (FPR), aka 1−specificity, where:

$$TPR = \frac{TP}{TP+FN}, \quad FPR = \frac{FP}{FP+TN}$$

To calculate the ROC curve for a given IIIC pattern, for example seizure (SZ), we convert expert votes from 6 classes into 2: SZ_+ = at least 50% of experts vote that the correct EEG segment label is seizure; SZ_- = less than 50% of experts vote that the correct EEG segment label is a seizure. *SzNet* outputs a probability value between 0 and 1 for each EEG segment. To compute the ROC curve for the SZ class, we compare *SzNet*'s probabilities for

all EEG segments in the test dataset with a threshold value θ, to obtain a set of binary classification decisions that depend on the threshold value. Relative to the ground truth labels (consensus label among the 20 experts), we can then calculate the model's false and true positive rates, $FPR(\theta)$, $TPR(\theta)$, which are the x and y coordinates of the ROC curve, respectively.

Operating points on the ROC plot for the experts are calculated similarly, with an important difference: When computing FPR and TPR for a given expert, the "ground truth" is calculated based on the *other* 19 experts' labels; we exclude the given expert's labels when calculating this ground truth ("*leave-one-expert-out*").

PR curves and expert operating points:
The PR curve (**Figure 1B**) shows the relationship between *SzNet*'s true positive rate (TPR) and its precision, aka positive predictive value (PPV), which is the fraction of positive cases predicted by the model that are in fact positive relative to the ground truth. TPR and PPV are defined as:

$$TPR = \frac{TP}{TP+FN}, \quad PPV = \frac{TP}{TP+FP}$$

To calculate the PR curve for a given IIIC pattern, for example LPD, we convert the 6-class classifications provided by experts into binary labels: LPD_+ = at least 50% of experts vote that the correct EEG segment label is LPD; LPD_- = less than 50% of experts vote that the correct EEG segment label is LPD. As above, we then compare *SzNet's* probabilities for all EEG segments in the test data set with a threshold value θ, to obtain a set of binary classification decisions that depend on the threshold value. Relative to the ground truth labels (consensus label among the 20 experts), we can then calculate the model's true positive rates and precisions, $TPR(\theta)$, $PPV(\theta)$, which are the x and y coordinates of the PR curve, respectively.

Operating points on the PR curve plot for the experts are calculated similarly, with the difference that, when computing PPV and TPR a given expert, the "ground truth" is calculated based on the *other* 19 experts' labels; we exclude the given expert's labels when calculating this ground truth ("*leave-one-expert-out*").

SzNet "outperforms" a given expert if that expert's operating point is below *SzNet*'s ROC or PR curve. This is because *SzNet* can be made to have any operating point along its ROC or PR curve by selecting the threshold θ. Thus, if a given expert's operating point on the ROC curve is $(FPR, TPR) = (a, b)$, then if *SzNet*'s ROC curve lies above this point, we can always select a threshold θ such that *SzNet* will have an operating point $(FPR, TPR) = (a, b')$, where $b' > b$, i.e., the same FPR, but a better sensitivity. Similar considerations apply to PR curves.

To provide an overall comparison between *SzNet* and the entire group of 20 experts, we report the percentage of the 20 experts that are outperformed by *SzNet,* both with respect to ROC curves and PR curves.

S6. Calibration
We use a parametric approach to evaluate experts' and *SzNet*'s statistical calibration. Results are shown in **Figure 1C** of the main text. The method for measuring expert calibration is described in the companion paper.

To obtain a comparable calibration curve for *SzNet*, we need a method to convert the probabilities the *SzNet* assigns to each EEG segment into final classification decisions. That is, we need to specify a rule by which to convert the 6 probabilities ("soft labels") into a final single decision ("hard label").

We consider two procedures to convert model probabilities ("soft labels") into definite classification ("hard label"). We call these the *max classification,* and *calibrated classification* procedures.

First-best classification: Assign the label that has maximum probability according to *SzNet*. For example, if the probabilities for a given EEG segment assigned by *SzNet* are:

[Pr(SZ), Pr(LPD), Pr(GPD), Pr(LRDA), Pr(GRDA), Pr(Other)] = [40, 35, 25, 10, 0, 0]%,

then we classify the segment as SZ. This "first best" rule performs produces classifications that are superior to most experts for most IIIC classes but was not well-calibrated for the "Other" class (results not shown). Therefore, we considered a more flexible, "first or second best" classification rule.

First or second-best classification: Classify based on the maximum probability, *unless* the difference between the two classes with highest and second highest probability is less than some specified threshold. One example of this rule that could apply when SZ is the maximum-probability class and LPD has the second highest probability is:

Classify as SZ, except when $\Pr(SZ) - \Pr(LPD) < \theta_{1,2}$. *In that case, classify as LPD*

where we are using 1 and 2 as indices for SZ and LPD. Such a rule may be adopted by an expert, for example, if the expert knows that two IIIC classes tend to be confused or to blend together, and / or if they consider it important to be more sensitive (higher TPR) for one class than the another. In general, this *first or second-best* classification rule can involve 30 different thresholds, one for each ordered pair among the 6 IIIC patterns. However, we reasoned that some second-best classifications should be excluded, and we thus assigned some of the thresholds *a priori* to zero. The form for the second-best classification threshold matrix that we adopted is shown in **Figure S7**:

Figure S7. Thresholds used for "First or second-best classification".

Second highest probability

Highest probability	SZ	LPD	GPD	LRDA	GRDA	Other
SZ	0	$\theta_{1,2}$	$\theta_{1,3}$	$\theta_{1,4}$	0	0
LPD	$\theta_{2,1}$	0	$\theta_{2,3}$	$\theta_{2,4}$	0	0
GPD	$\theta_{3,1}$	$\theta_{3,2}$	0	0	$\theta_{3,5}$	0
LRDA	$\theta_{4,1}$	$\theta_{4,2}$	0	0	$\theta_{4,5}$	0
GRDA	$\theta_{5,1}$	0	$\theta_{5,3}$	$\theta_{5,4}$	0	$\theta_{5,6}$
Other	$\theta_{6,1}$	$\theta_{6,2}$	$\theta_{6,3}$	$\theta_{6,4}$	$\theta_{6,5}$	0

The rationale for each row (corresponding to the highest probability class) is:

- SZ: The borders are blurry between this pattern and LPD, GPD, LRDA, thus it is reasonable to classify a pattern as the one with second highest probability in theses.
- LPD: The borders are blurry between this pattern and SZ, GPD, LRDA.
- GPD: The borders are blurry between this pattern and SZ, LPD, GRDA.

- LRDA: The borders are blurry between this pattern and SZ, LPD, GRDA.
- GRDA: The borders are blurry between this pattern and SZ, GPD, LRDA, and Other.
- Other: The borders are blurry between this and all other patterns.

To obtain values for the thresholds in the above matrix that improve model calibration relative to experts, we performed 10,000 rounds of a simple random search: In each round, random values for the non-zero entries in the matrix were generated by sampling from a uniform [0, 1] distribution. For each random matrix, we calculated the majority and pairwise inter-rater reliability (mIRR) between *SzNet* and the 20 experts in the *training data;* no test data was used in this procedure. Calculations for mIRR and pIRR are explained in the companion paper. After 10,000 rounds of random sampling, we selected the decision threshold matrix that yielded the minimum average difference of mIRR between *SzNet* and experts on the training data.

This "first or second-best classification" rule learned using the training data was subsequently used to derive model classification decisions for each EEG segment. These classifications were used to evaluate *SzNet*'s statistical calibration on the test dataset. Results are shown in Figure 1C.

S7. Pairwise and majority reliability

Performance metrics

We calculate pairwise and majority inter-rater reliability metrics (mIRR, pIRR) (**Figure S8 A, B**). The methods for these calculations are adapted from Supplemental Material S4 in the companion paper. We calculate 4 IRR metrics, shown in:

- ee-mIRR: Majority confusion matrix, comparing experts to experts (ee).
- ea-mIRR: Majority confusion matrix, comparing algorithm to experts (ea)
- ee-pIRR: Pairwise confusion matrix, comparing experts to experts (ee).
- ea-pIRR: Pairwise confusion matrix, comparing algorithm to experts (ea)

Calculations for ee-mIRR and ee-pIRR are described in the companion paper. The same methods are adapted to calculate ea-mIRR and ea-pIRR, with the following differences:

- ea-mIRR quantifies the agreement of *SzNet* with the expert majority label (the consensus among the labels from the 20 experts), whereas ee-mIRR is computed by calculating each expert's agreement with the majority / consensus label of the remaining 19 experts and then taking the average across all 20 experts.
- ea-pIRR quantifies the average pairwise agreement of *SzNet* with each expert, where each expert and *SzNet* take turns standing in as the "gold standard". For *SzNet,* this quantity represents an average over 40 pairs (first 20: *SzNet* as gold standard, and expert $i = 1, 2, \ldots, 20$ as the comparator; second 20: each expert $i = 1, 2, \ldots, 20$ as the gold standard, and *SzNet* as the comparator). By contrast, ee-pIRR is an average over 380 ordered pairs of experts.

Note that ea-mIRR and ea-pIRR are the values along the diagonals of the confusion matrices, ea-mCM and ea-pCM, respectively. Similarly, ee-mIRR and ee-pIRR are the values along the diagonals of the confusion matrices ee-mCM and ee-pCM.

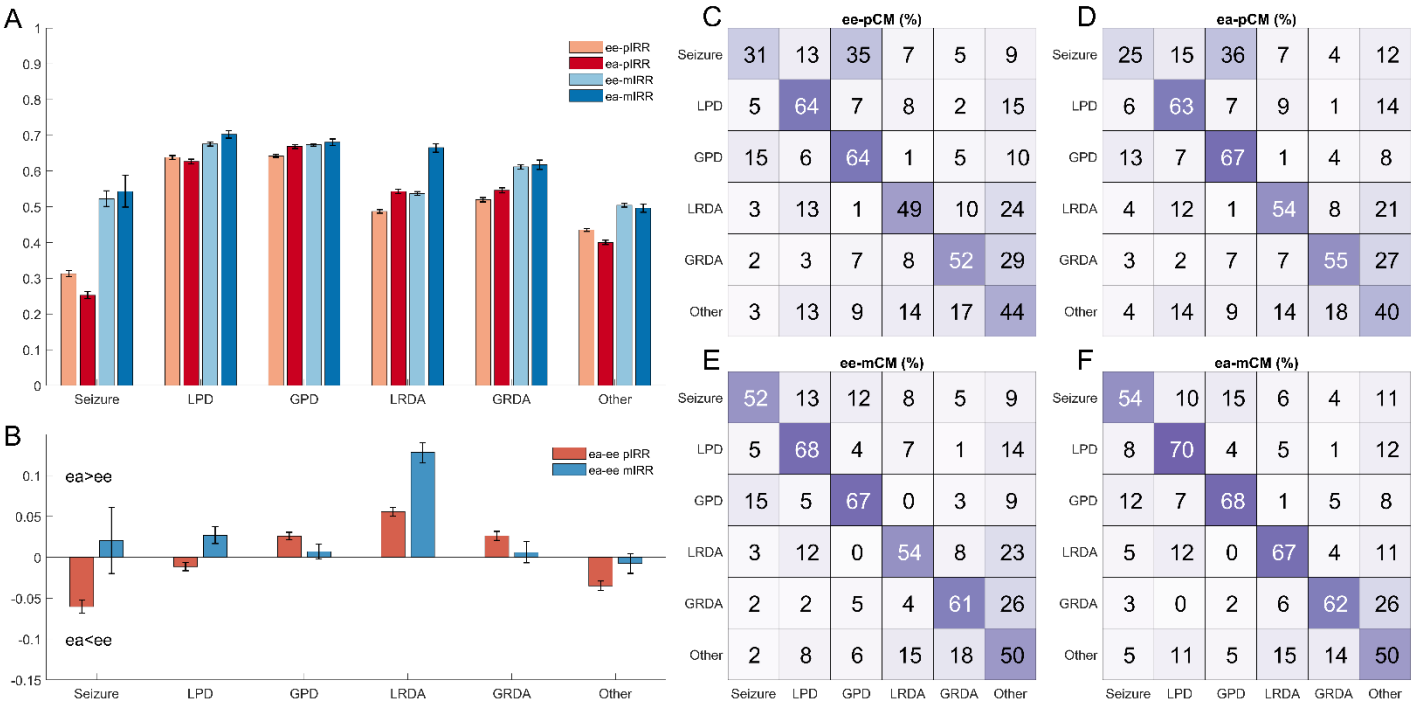
S8. Confusion matrices

We calculate pairwise and majority confusion matrices (mCM, pCM, **Figure S8, C, D, E, F**) using methods adapted from Supplemental Material S6 in the companion paper. We define 4 types of confusion matrices:

- ee-mCM: Majority confusion matrix, comparing experts to experts (ee).
- ea-mCM: Majority confusion matrix, comparing algorithm to experts (ea)
- ee-pCM: Pairwise confusion matrix, comparing experts to experts (ee).

- ea-pCM: Pairwise confusion matrix, comparing algorithm to experts (ea)

Figure S8. Additional performance metrics for *SzNet*. (A) Bar plots showing average inter-rater reliability (IRR) between pairs of experts (ee-pIRR; light red bars) and the average agreement between the algorithm and each expert (ea-pIRR; dark red bars); and average agreement of experts with the label assigned by the majority of other experts (ee-mIRR) and of the algorithm with the majority of experts (ea-mIRR). The differences for each of these pairs is shown in B, with values above 0 indicating better performance for the algorithm (ea>ee), and values below 0 indicating better performance among experts. Confidence intervals are calculated via bootstrapping. The confusion matrices (CM) in subplots C, D, E, F expand on the IRR results, showing not only how well experts and algorithm agree with the label being treated for each analysis as the "correct answer" (numbers along the diagonals), but also showing the distribution of disagreements (values along each row).



S9. Embedding maps

To visualize the relationships between IIIC patterns learned by *SzNet,* we employed a dimensionality reduction method. The dimensionality is reduced from the 6-deminsional model output vector concatenated with the total IIIC (combined class of seizure, L/GPD, and LRDA) score as a 7th dimension, using a set of 180K labeled EEG segments balance-sampled from our EEG cohort. We further color the UMAP with various schemes including expert consensus labels, class inferred by the model, model uncertainty, seizure probability, and total IIIC probability.

Hyper-parameters for the UMAP algorithm were set as follows[8]. The dimensions of the output map were set to (2,1), to allow plotting in 2D. The size of the local neighborhood was set 15. The minimum distance apart that EEG segments can be in the low dimensional representation was set to 0.1. The distance metric was set to be Euclidean Distance. The initial embedding position assignments were set to be random. The resultant embedding maps are shown in Figure 2 of the main text.

S10. Literature review

To assess the current state of the art on automated classification of seizures and other IIIC patterns, we performed a systematic literature review in PubMed. The search query was the following: "(EEG OR electroencephalogram*) AND (autom* OR comput* OR machine) AND (seizure* OR inter-ictal OR interictal) AND (classif* OR detect* OR identif*) NOT intracranial", where the wildcards (*) were used to look for all possible endings to the root terms in the database. The Preferred Reporting Items for Systematic Reviews and

Meta-Analyses (PRISMA) with inclusion and exclusion criteria for article selection is depicted in Fig. S9. The records retrieved from PubMed by this search were subjected to expert screening. From the collection of eligible articles, 19 met eligibility criteria and are summarized in Table S1. The remaining articles were excluded from the analysis since they did not fit the eligibility criteria, which consisted of:

Inclusion criteria

- Subjects include adults (age ≥18)
- Describe development and/or testing of algorithms for seizure and /or IIIC pattern classification
- Include at least 50 subjects

Exclusion criteria

- Exclusively about neonates or children
- Intracranial EEG

Several conclusions can be drawn from **Table S1** and **Figure S9** about most published studies in the field:

- Most involve very small datasets.
- Most either involve small labels from very numbers of clinical experts, or do not report how labels were generated.
- Most focus exclusively on seizure detection and do not attempt to classify other types of IIIC patterns.

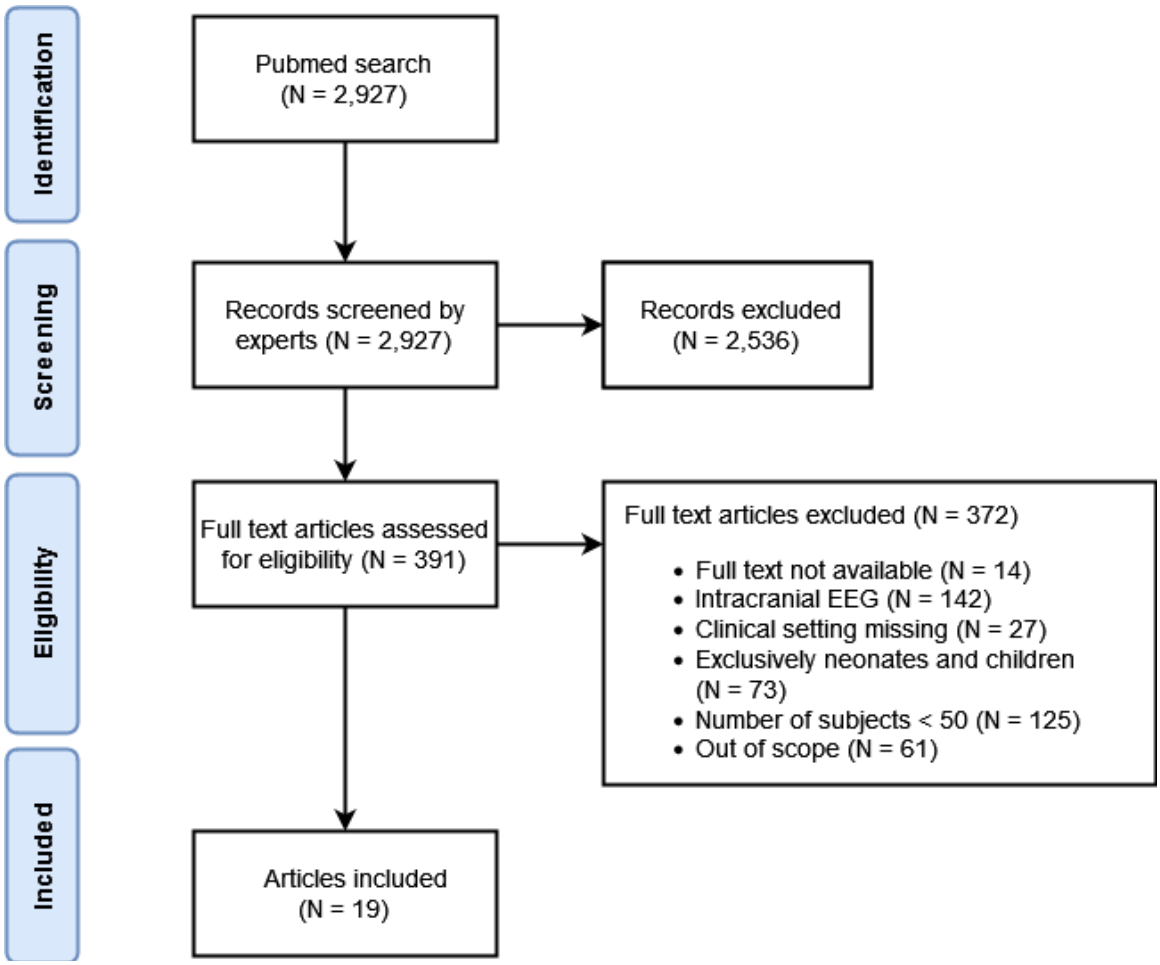
Table S1. Prior literature in seizures and IIIC patterns detection.

(Year)[ref]	Setting	Event type	Algorithm type	No. subjects (type)	# Raters (type)	Type of Study
(2021) [9]	EMU	Seizures	Commercial software.	120 (adults)	3 CNPP	Algorithm evaluation.
(2021) [10]	EMU, ICU, NICU	Seizures	Deep neural network.	365 (neonates, children, adults)	Not reported	Algorithm development and evaluation.
(2021) [11]	ED, ICU	Seizures	Commercial software.	353 (adults)	2 CNPP	Algorithm evaluation.
(2021) [12]	EMU, ICU, NICU	Seizures	Deep neural network.	615 (neonates, children, adults)	Not reported	Algorithm development and evaluation.
(2021) [13]	ICU	Seizures	Random forest.	97 (adults)	Not reported.	Algorithm development and evaluation.
(2020) [14]	ICU, PICU	Seizures	CNN.	498 (children and adults) + 5067 “weak labels”	1 CNPP	Algorithm development and evaluation.
(2020) [15]	EMU, ICU, NICU	Seizures	Dynamic Bayesian modeling.	131 (neonates, children, adults)	Not reported	Algorithm development and evaluation.
(2020) [16]	EMU, ICU	Seizures	Deep Neural Network.	78 (children and adults)	Not reported	Algorithm development and evaluation.
(2020) [17]	EMU, ICU, NICU	Seizures	Deep Neural Network.	246 (neonates, children, adults)	Not reported.	Algorithm development and evaluation.
(2020) [18]	EMU, ICU, NICU	Absence seizures	Unsupervised learning.	637 (neonates, children, adults)	Not reported.	Algorithm development and evaluation.
(2019) [19]	EMU, ICU, NICU	Seizures	Extreme gradient boosting.	114 (neonates, children, adults)	Not reported.	Algorithm development and evaluation.
(2019) [20]	Ambulatory	Seizures	Commercial software.	70 (adults)	2 CNPP.	Algorithm validation.
(2017) [21]	EMU	Seizures	Commercial software.	92 (adults)	1 CNPP	Algorithm evaluation.
(2016) [22]	ICU, NICU	Seizures	Support vector machines.	78 (neonates and adults)	Not reported.	Algorithm development and evaluation.
(2016) [23]	ICU	Seizures	Support vector machines.	53 (adults)	Not reported.	Algorithm development and evaluation.
(2015) [24]	ICU	Seizures	Commercial software.	98 (children and adults)	2 CNPP.	Algorithm evaluation.

(2015) [25]	EMU	Seizures	Commercial software.	515 (adults)	Not reported	Algorithm evaluation
(2015) [26]	ICU	Seizures, IIIC	Commercial software.	68 (adults)	2 CNPP	Algorithm evaluation
(2010) [27]	EMU	Seizures	Commercial software.	102 (adults)	3 CNPP.	Algorithm evaluation.

ICU – Intensive care unit; EMU – Epilepsy monitoring unit; ED – Emergency Department; NICU – Neonatal ICU; PICU – Pediatric ICU; CNPP –clinical neurophysiologist physicians.

Figure S9. PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) flow diagram, where N corresponds to the number of articles.



REFERENCES

1. Bagheri, E. *et al.* Interictal epileptiform discharge characteristics underlying expert interrater agreement. *Clin. Neurophysiol. Off. J. Int. Fed. Clin. Neurophysiol.* **128**, 1994–2005 (2017).

2. Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. Densely connected convolutional networks. in 4700–4708 (2017).

3. Bengio, Y., Goodfellow, I. & Courville, A. *Deep learning*. vol. 1 (MIT press Massachusetts, USA:, 2017).

4. Kullback, S. *Information theory and statistics*. (Courier Corporation, 1997).

5. Kullback, S. & Leibler, R. A. On information and sufficiency. *Ann. Math. Stat.* **22**, 79–86 (1951).

6. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. *ArXiv14126980 Cs* (2017).

7. McInnes, L., Healy, J. & Melville, J. Umap: uniform manifold approximation and projection for dimension reduction.[Google Scholar]. (2018).

8. https://umap-learn.readthedocs.io/en/latest/.

9. Scheuer, M. L. *et al.* Seizure Detection: Interreader Agreement and Detection Algorithm Assessments Using a Large Dataset. *J. Clin. Neurophysiol. Off. Publ. Am. Electroencephalogr. Soc.* **38**, 439–447 (2021).

10. Roy, S. *et al.* Evaluation of artificial intelligence systems for assisting neurologists with fast and accurate annotations of scalp electroencephalography data. *EBioMedicine* **66**, 103275 (2021).

11. Kamousi, B. *et al.* Monitoring the Burden of Seizures and Highly Epileptiform Patterns in Critical Care with a Novel Machine Learning Method. *Neurocrit. Care* **34**, 908–917 (2021).

12. Cao, X., Yao, B., Chen, B., Sun, W. & Tan, G. Automatic Seizure Classification Based on Domain-Invariant Deep Representation of EEG. *Front. Neurosci.* **15**, 760987 (2021).

13. Bernabei, J. M. *et al.* A Full-Stack Application for Detecting Seizures and Reducing Data During Continuous Electroencephalogram Monitoring. *Crit. Care Explor.* **3**, e0476 (2021).

14. Saab, K., Dunnmon, J., Ré, C., Rubin, D. & Lee-Messer, C. Weak supervision as an efficient approach for automated seizure detection in electroencephalography. *NPJ Digit. Med.* **3**, 59 (2020).

15. Song, X., Aguilar, L. & Yoon, S.-C. A Comparison of Dynamic Modeling Approaches for Epileptic EEG Detection and Classification. *Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Int. Conf.* **2020**, 523–527 (2020).

16. Ayodele, K. P., Ikezogwo, W. O., Komolafe, M. A. & Ogunbona, P. Supervised domain generalization for integration of disparate scalp EEG datasets for automatic epileptic seizure detection. *Comput. Biol. Med.* **120**, 103757 (2020).

17. Iešmantas, T. & Alzbutas, R. Convolutional neural network for detection and classification of seizures in clinical data. *Med. Biol. Eng. Comput.* **58**, 1919–1932 (2020).

18. Tsiouris, K. M., Konitsiotis, S., Gatsios, D., Koutsouris, D. D. & Fotiadis, D. I. Automatic Absence Seizures Detection in EEG signals: An Unsupervised Module. *Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Int. Conf.* **2020**, 532–535 (2020).

19. Vanabelle, P., De Handschutter, P., El Tahry, R., Benjelloun, M. & Boukhebouze, M. Epileptic seizure detection using EEG signals and extreme gradient boosting. *J. Biomed. Res.* **34**, 228–239 (2019).

20. González Otárula, K. A., Mikhaeil-Demo, Y., Bachman, E. M., Balaguera, P. & Schuele, S. Automated seizure detection accuracy for ambulatory EEG recordings. *Neurology* **92**, e1540–e1546 (2019).

21. Fürbass, F. *et al.* Automatic multimodal detection for long-term seizure documentation in epilepsy. *Clin. Neurophysiol. Off. J. Int. Fed. Clin. Neurophysiol.* **128**, 1466–1472 (2017).

22. Bogaarts, J. G., Gommer, E. D., Hilkman, D. M. W., van Kranen-Mastenbroek, V. H. J. M. & Reulen, J. P. H. Optimal training dataset composition for SVM-based, age-independent, automated epileptic seizure detection. *Med. Biol. Eng. Comput.* **54**, 1285–1293 (2016).

23. Bogaarts, J. G., Hilkman, D. M. W., Gommer, E. D., van Kranen-Mastenbroek, V. H. J. M. & Reulen, J. P. H. Improved epileptic seizure detection combining dynamic feature normalization with EEG novelty detection. *Med. Biol. Eng. Comput.* **54**, 1883–1892 (2016).

24. Sierra-Marcos, A., Scheuer, M. L. & Rossetti, A. O. Seizure detection with automated EEG analysis: a validation study focusing on periodic patterns. *Clin. Neurophysiol.* **126**, 456–462 (2015).

25. Fürbass, F. *et al.* Prospective multi-center study of an automatic online seizure detection system for epilepsy monitoring units. *Clin. Neurophysiol. Off. J. Int. Fed. Clin. Neurophysiol.* **126**, 1124–1131 (2015).

26. Herta, J. *et al.* Prospective assessment and validation of rhythmic and periodic pattern detection in NeuroTrend: A new approach for screening continuous EEG in the intensive care unit. *Epilepsy Behav. EB* **49**, 273–279 (2015).

27. Kelly, K. M. *et al.* Assessment of a scalp EEG-based automated seizure detection system. *Clin. Neurophysiol. Off. J. Int. Fed. Clin. Neurophysiol.* **121**, 1832–1843 (2010).