Avoiding Post-processing with Event-Based Detection in Biomedical Signals

Nick Seeuws, Maarten De Vos, and Alexander Bertrand

***Abstract*— *Objective:* Finding events of interest is a common task in biomedical signal processing. The detection of epileptic seizures and signal artefacts are two key examples. Epoch-based classification is the typical machine learning framework to detect such signal events because of the straightforward application of classical machine learning techniques. Usually, post-processing is required to achieve good performance and enforce temporal dependencies. Designing the right post-processing scheme to convert these classification outputs into events is a tedious, and labor-intensive element of this framework. *Methods:* We propose an event-based modeling framework that directly works with events as learning targets, stepping away from ad-hoc post-processing schemes to turn model outputs into events. We illustrate the practical power of this framework on simulated data and real-world data, comparing it to epoch-based modeling approaches. *Results:* We show that event-based modeling (without tailored post-processing) performs on par with or better than epoch-based modeling with extensive post-processing. *Conclusion:* These results show the power of treating events as direct learning targets, instead of using ad-hoc post-processing to obtain them, severely reducing design effort. *Significance* The event-based modeling framework can easily be applied to other event detection problems in signal processing, removing the need for intensive task-specific post-processing.**

***Index Terms*— Biomedical Signal Processing, Deep Learning, Neural Networks**

I. Introduction

Machine learning has become a popular approach to solve biomedical signal processing problems, for example for epileptic seizure detection [1]–[4], the detection of sleep events [5]–[12] and sleep stages [13], and the detection of signal disturbances, also known as signal artefacts [14]–[19].

A specific group of tasks in biomedical signal processing can be said to deal with *events*. Conceptually, these tasks involve the detection of transitions to a signal event of interest and back to the background signal. The detection of epileptic seizures and signal artefacts are two such examples, where the seizures and artefacts are the events of interest. The manual event detection process involves scrolling through signal recordings (potentially representing multiple hours of signal) and logging the start and stop times of the events. Designing tools to aid human annotators requires linking raw signal recordings with a set of (t_{start}, t_{stop}) tuples.

The prototypical machine learning approach to detect such events in biomedical signals involves segmenting a signal recording into distinct epochs and predicting a label for each individual epoch [20]. Deciding on the epoch duration is part of the design process, and also heavily relies on the machine learning model that is used. An extreme case is where a prediction is made on the level of individual time samples (which could be viewed as single-sample epochs), as often done in U-Net-like architectures [3], [5].

Converting predictions for every individual epoch into a (t_{start}, t_{stop}) tuple spanning a continuous event, possibly across multiple epochs, involves extensive post-processing [1]–[3]. This post-processing is usually not learned from data, but designed based on expert knowledge. If epochs are fed to a machine learning model independently the post-processing stage is also responsible for encoding temporal dependencies inherent to time series processing into the final output. This responsibility makes the post-processing stage a crucial ingredient with a tedious design process.

In this work, we introduce *event-based* modeling as an alternative paradigm (instead of epoch-based modeling). We will illustrate how this method bypasses the need for a tedious ad-hoc design of a proper post-processing stage. Inspired by works in visual object detection, our method encodes events of interest using the events' center and duration. Both are predicted jointly by a single deep learning model. We consider this to be an *event-based* approach to biomedical signal analysis.

Event-based modeling reduces many separate design tasks to the design of a single neural network, without a need to carefully tune pre- or post-processing steps (as both are directly learned by the model). Encoding training events involves mapping the different events to an *event center* and *duration* signal, after which the model can be trained end-to-end without post-processing. Explicitly modeling events, combined with the end-to-end nature of deep learning, encourages the model to properly learn the full character and diversity of target events. Crucially, we can easily cope with a large

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 802895) and from the Flemish Government (AI Research Program).

N. Seeuws , A. Bertrand and M. De Vos are with the Dept. of Electrical Engineering (ESAT), Stadius Center for Dynamical Systems, Signal Processing and Data Analytics (STADIUS), KU Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium

M. De Vos is also with the Dept. of Development and Regeneration, Faculty of Medicine, KU Leuven

A. Bertrand and N. Seeuws are affiliated to Leuven.AI - KU Leuven institute for AI, B-3000, Leuven, Belgium.

variability in event duration.

To summarize, our key contributions are as follows:

- Introduction of a generic *event-based* modeling framework for biomedical signal processing,
- An event-detection algorithm that does not require tailored, task-specific post-processing (in contrast to most epoch-based approaches),
- Due to its end-to-end nature, our algorithm learns the full character and diversity of target events of variable duration.

Section II introduces our event-based framework. Section III discusses our experiments and illustrates the performance of the proposed framework on synthetic and biophysical data. Section IV discusses these results and explains benefits and drawbacks of using an event-based framework. Section V concludes the paper.

II. Methods

A. Event-based modeling

In this subsection, we describe the event-based framework conceptually, and we refer to Figure 1 for a schematic illustration of the different concepts introduced in this subsection.

1) Encoding and decoding events: Our event-based framework represents events by their center point and duration. A signal recording is used as input, and the goal is to produce center and duration predictions across the length of the recording. Both outputs (i.e., center and duration) are treated as "signals", in the sense that they span the entire input. This situates our approach in the area of "sequence to sequence" learning, similar to the works of [3], [5], [13]. The *center signal* indicates whether a point in time corresponds to the center of an event. The *duration signal* is used to represent an event's duration *if that point in time would be a center point*. Note that this duration signal is meaningless at time points far away from a center point. Our approach is inspired by CenterNet [21] for object detection in images. This image object detection model predicts centers of detected objects, and predicts object sizes at those specific centers.

At inference time, the center and duration outputs are decoded by searching for the peaks in the center signal. The detected event centers are then represented by the different peaks, and the confidence level for each event is displayed by the specific signal values at the corresponding peaks. [1] For each detected center, the predicted event duration can then be found in the duration signal at that specific point.

In our experiments, we locate the center signal peaks by listing all local maxima (that exceed a certain threshold) using a peak finding routine of SciPy [22]. Only relying on local maxima, however, can lead to center points that are close to one another (and lead to overlapping event predictions.) If this is the case, there might be a need for merging these overlapping events, or non-maximum suppression (as is common in computer vision [21], [23]–[26]). For our experiments on epileptic seizures (explained in section III-C.0.b), we use non-maximum suppression for events that have more than 0.5 Intersection-over-Union to avoid occasional double detections. Note that these are *generic* post-processing schemes.

2) Losses:

a) Training targets: To train the center and duration predictions, training targets need to be defined. Center prediction is treated as a sample-based classification task, similar to the approaches of [3], [5]. In contrast to most sample-based classification approaches, some slack should be allowed on the center targets. Predicting a center that is just a few samples off-target is better than, e.g., predicting a false event in an hour of background signal, and thus should be penalized less. The weighting method of [21] for object detection is modified and applied to event center prediction. For an event with ground truth center t^*, the target center signal is defined as

$$c(t) = \exp(-\frac{(t-t^*)^2}{2\sigma^2}) \tag{1}$$

with σ depending on the target event's duration. Following [23], this hyperparameter is set as $\sigma = d/12$, with d the event's actual duration, measured in terms of *time points*. The hyperparameter can be adjusted for a specific use case to specify how precise a model should be in its center predictions during training (e.g, a larger value for σ allows for more "slack" on center predictions.) In this work, we use the values of [23] as-is, and leave tuning of the hyperparameter as future work. In the case of multiple events in a signal, these center target signals as in Eq. 1 are defined for each event independently, and are combined by taking the maximum target value at every time point.

The duration targets only need to be defined at the target centers t^*, since only duration predictions at ground-truth center points t^* will be considered in the duration loss. As presented, we chose to work with a maximum duration that the model can predict. This can either be set to the maximum duration in a given data set, or determined with expert information (e.g.,*"What can reasonably be expected as an upper bound for these events?"*). In principle, the duration prediction can be unbounded (one will not always know the maximal duration a priori), but constraining the duration prediction leads to improved stability during training. The duration predictions, similar to the center predictions, are constrained to the range $[0, 1]$ (the target durations are divided by the predefined maximum duration.) For every target event in a data set, the duration signal value at the event's center point is set to the event's normalized duration.

b) Training losses: Center prediction is trained using focal loss [24], in the modified form of [25]. The full center signal prediction loss L_c for an input signal containing N events, center prediction $c'(t)$, and target center signal $c(t)$ is defined as:

$$L_c = -\frac{1}{N}\sum_t \begin{cases} (1-a_c)(1-c')^\alpha \log(c') & \text{if } c = 1 \\ a_c(1-c)^\beta c'^\alpha \log(1-c') & \text{otherwise} \end{cases} \tag{2}$$

[1] This confidence is expressed on a relative scale, and does not necessarily represent detection probabilities like they do for classification problems. These signal values should be interpreted as "confidence scores", e.g., a model is more confident in its prediction for an event with score 0.8 than it is for an event with 0.3 as score.

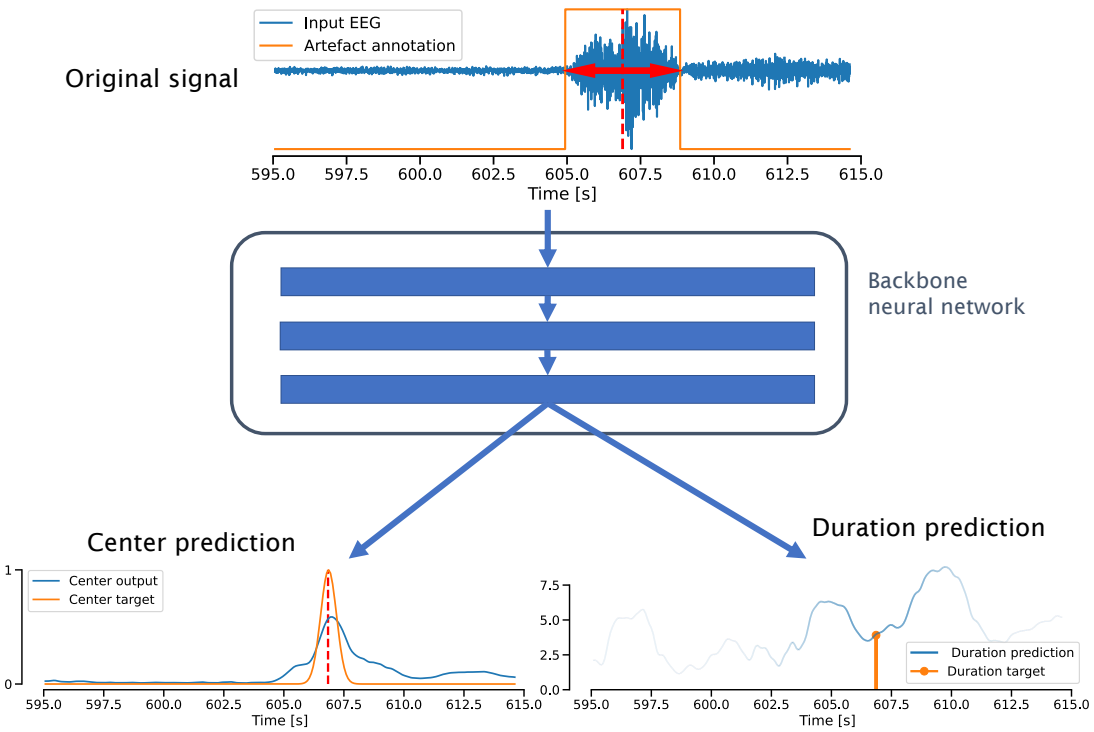


Fig. 1: Event-based modeling overview. The input EEG signal at the top contains a single artefact event, annotated in orange (the event's center and duration are highlighted in red). The neural backbone can be any sequence-to-sequence model. Two independent convolutional layers output the *center* and *duration* signals (in blue). The training targets for both signals are plotted in orange. Training the *center* signal involves comparing it to the entire target signal. The *duration* signal is only trained and evaluated at event centers. Note that this example only spans 20 s and contains a single, easy to detect event. With the right backbone, the approach can process longer inputs and detect more events at once.

(with the dependence on t of $c'(t)$ and $c(t)$ dropped for legibility). Hyperparameters are set as $a_c = 0.1, \alpha = 2$ and $\beta = 4$ following [24], [25]. The original focal loss is a classification loss where mistakes are adaptively penalized based on the model's confidence using the factors exponentiated with α. With the modification of [25], false alarms close in time to the target center t^* are penalized less than false alarms further away (using the factor $(1-c)^\beta$) and the exponential in Eq. 1). [2]

Duration predictions are trained using *Intersection over Union* (IoU) as loss. IoU is a popular loss formulation in object detection [23]. Crucially, IoU is based on *relative* duration errors, ensuring that the batch loss will not be dominated by long events. Calculating the intersection and union of predicted and target events can be simplified due to the use of the target event's center. The duration loss L_d, for N events, set of known target points $\mathcal{T}$, predicted duration signal $d'(t)$ and target $d(t)$, is formulated as

$$L_d = \frac{1}{N} \sum_{t^* \in \mathcal{T}} \frac{\min[d'(t^*), d(t^*)]}{\max[d'(t^*), d(t^*)]} \tag{3}$$

Note that the predicted duration signal $d'(t)$ is only evaluated at the center points, i.e., the value of $d(t)$ and $d'(t)$ has no meaning at points which are not treated as center points, even though the network will produce an output $d'(t)$ for every time point t.

The center prediction loss L_c and duration loss L_d are combined into the full loss L as a weighted sum,

$$L = L_c + \lambda_d L_d$$

, where λ_d is a hyperparameter to control the relative influence of the two tasks (center and duration prediction). In our experiments, both loss terms become approximately equal in magnitude by setting $\lambda_d = 5$. This value can be raised or lowered to increase or decrease, respectively, the influence of the duration prediction task.

3) Backbone model: Taking an event-based approach to modeling events does not rely on a specific backbone architecture. As discussed above, one can view it as a specific instance of sequence-to-sequence modeling. Hence, any neural network architecture that maps an input signal to an output signal can be applied in an event-based context (one would need to account for the center and duration signals by converting the architecture to produce two outputs).

In our experiments, we use U-Net-like backbones (tailored to a specific data set), ensuring we use the same backbone architecture for the event-based and epoch-based approaches for fair comparison. This type of architecture won two recent

[2]This form of the focal loss can cause numerical instabilities when performing gradient descent during training. To avoid instabilities, one needs to rewrite the loss in terms of the logits instead of sigmoid inputs. Refer to the Supplementary Material for more details.

machine learning competitions in the context of event detection in biomedical signals[3], [5]. The architecture is capable of mapping an input signal (uni- or multivariate) to the desired center and duration output signals, and manages to combine global and local information of the input. Due to differences in scale of the target events in our experiments, discussed below, specific implementations of the backbones are tailored to the specific data sets. Details can be found in the Supplementary Material.

Our model implementations, experiment code and Supplementary Material can be found at `https://github.com/nseeuws/EventBasedModeling`

III. Experiments and Results

A. Measuring performance

Measuring performance of an algorithm in the context of biomedical events is not straightforward. In the field of epileptic seizure detection, for example, multiple measures are in active use, and all focus on different aspects of "performance" [27]. Broadly speaking, there are two categories of performance measures: epoch-based and event-based. Epoch-based measures treat the *evaluation* of an algorithm similarly to epoch-based solutions, i.e., as separate classification for each epoch. In this case, one can rely on classical performance measures for classification problems (accuracy, precision, recall, etc.).

In the context of this work, however, we elect to measure performance using an event-based measure. Event-based measures take a more holistic approach to evaluation and focus on the events in question, not on epoch-centric classification results. Designing an *event-based* model is mainly relevant if the final evaluation will also take an event-based point of view, motivating our choice for this type of evaluation. Event-based measures look at how well predictions overlap with reference annotations, and match predicted events with reference events. Different measures vary in how they quantify overlap between predictions and references, and what they consider as "enough overlap". For example, in [27] the authors discuss, among others, the *any-overlap* and *time-aligned event scoring* methods. Broadly speaking, the former considers a prediction to be a correct prediction if there is any temporal overlap with ground-truth events, while the latter also considers the amount of overlap between predictions and ground truth. Both measures can give different results for a single set of predictions, and it is up to the user to decide what measure best corresponds to the problem at hand. In object detection for computer vision, *object-based* (the computer vision equivalent of event-based) evaluation is also common [21], [23]–[26], [28]. Matches between the set of predicted objects and ground-truth objects are made based on maximal overlap, measured using *Intersection-over-Union* (IoU). For the prediction-ground-truth pairs obtained like this, if the overlap is higher than a specific IoU threshold, the pair counts as a correct detection.

Throughout this paper, we will use the IoU as the main performance metric. Changing the IoU threshold for evaluation allows us to elegantly make evaluation more or less strict, depending on the problem at hand. As an illustration of the power of event-based modeling, evaluation flexibility is desirable in the context of this paper. For example, the *any-overlap* scoring of [27] (a lenient evaluation criterion) can be seen as a limit case of the IoU threshold going to zero, and will be applicable in use cases where the *any-overlap* scoring is also relevant. On the other hand, setting a high IoU threshold is suited to evaluate algorithms when there is a high standard on the overlap between predicted and ground-truth events.

B. Simulated events

As a first test we simulate a data set by mixing realistic noise events in electrocardiography (ECG) signals. This ensures unambiguous annotations, which can be difficult to obtain in real-life biomedical data (e.g., in epileptic seizure detection, where the precise starting point of seizure is difficult to define). As background signal, we use the Computing in Cardiology 2017 Challenge data set [29]. This data set contains lead I ECG recordings of sinus rhythm ECG and atrial fibrillation. We randomly generate electrode artefact events with varying durations sourced from the Physionet MIT-BIH Noise Stress Test Database [30]. The artefact events are added to the background signal with varying SNR levels. To "smoothen" the transition between background and artefact, the artefacts are elementwise multiplied with a Tukey window of the same size as the artefact event. To make the task more difficult, we add short bursts of artefact signal throughout the data set (which should be ignored by the models). Example events and corresponding predictions are visualized in Figure 2. Full data generation details are discussed in the Supplementary Material.

We compare our event-based approach to a generic epoch-based approach. Using a U-Net-like backbone for event-based modeling allows for direct comparison with epoch-based modeling by training an actual U-Net (with a single output) in an epoch-based manner, where it is trained to produce predictions at $1/16$ the original sampling rate (making it so that the two settings use the same backbone architecture and the addition of the center and duration output is the only change). Network details are explained in the Supplementary Material. The event-based model is used as-is, while the epoch-based model is used in three different settings:

- *No post-processing*: To establish a baseline, we report performance of the epoch-based approach *without* post-processing to gauge the impact of post-processing schemes.
- *Median filtering*: To compensate for potential (short) false positives, which we know to be a risk due to the short "distraction" events added into the data generation process, we use median filtering after thresholding the base-model output as a first post-processing scheme. The filter length is equivalent to $1\,\mathrm{s}$, the shortest possible duration of the target events.
- *Morphological operations*: As a more advanced post-processing scheme, we take inspiration from morphological operations popular in the field of computer vision [31]. After training the base epoch-based model, we observed more room for improving results than median

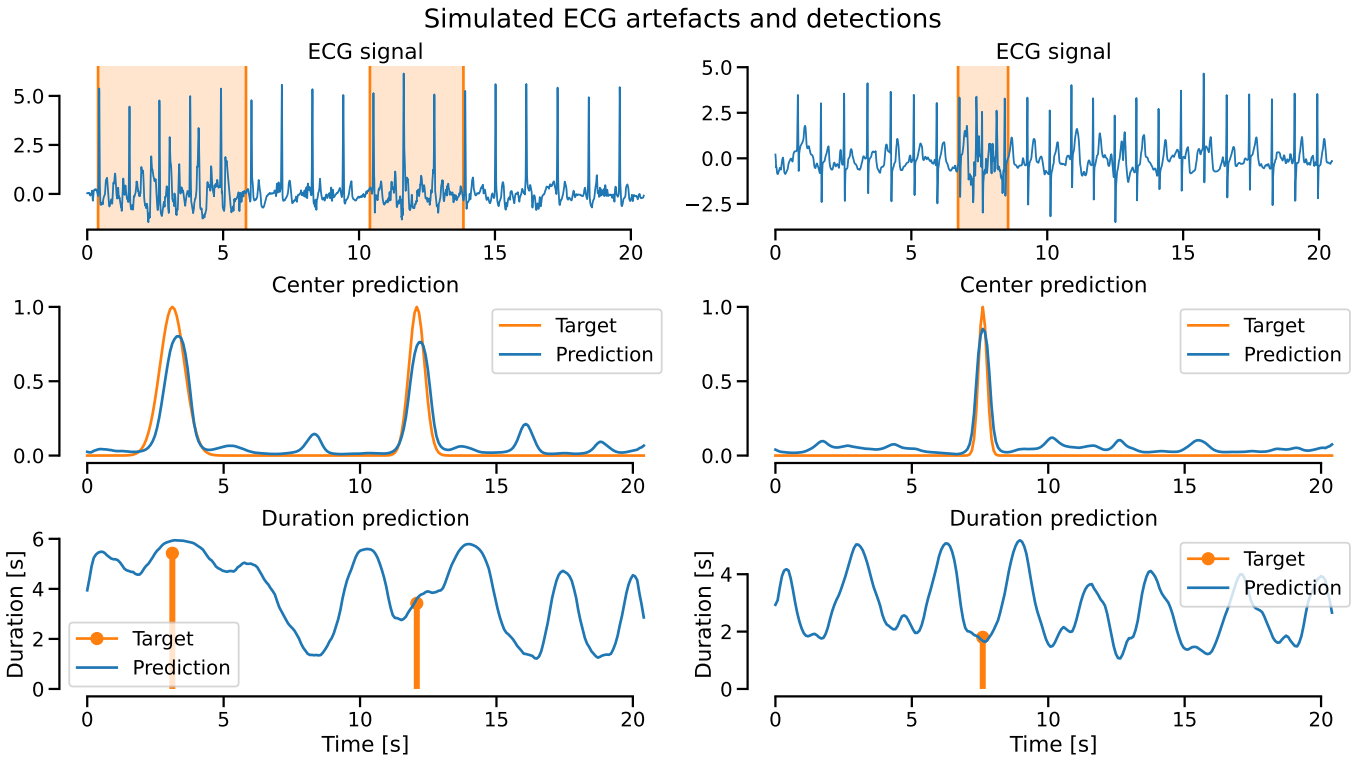


Fig. 2: Examples of simulated events and corresponding predictions. The vertical orange lines in the signal plots indicate the start and stop times of each target event. Signal amplitude is unitless in both subplots. Duration predictions and targets are rescaled to reflect duration in seconds.

filtering. The U-Net would predict "holes" in target events which are not always fixed with median filtering (especially if they occur close to event edges), in addition to false positives. To further push post-processing, we use *binary closing* (which closes holes in the foreground, i.e., a predicted event), followed by *binary opening* (which removes short events, expected to be false positives) with a binary structuring element of size equivalent to 1 s, applied after thresholding the base-model output.

To measure performance, we use two different IoU thresholds, 0.25 and 0.75, to determine "hits" and "misses" of the two approaches. These two thresholds represent two different evaluation settings, one where the overlap between predictions and ground truth is not that important (0.25 IoU) similar to the *any-overlap*, and a setting where the precise overlap is more important (0.75 IoU). We randomly generate 25 data sets to control for variability. We report the test set F1-score corresponding to the confidence threshold that gives the highest F1-score on the validation set. For the event-based approach, we include events with a confidence score higher than the threshold. Operating points for the epoch-based approaches are set by thresholding the model output by a certain value. We opt for the F1-score instead of looking at the full precision-recall curves since the epoch-based approaches are observed to only have a singular Pareto-optimal operating point (illustrated in the Supplementary Material). The epoch-based approaches would thus realistically only be used at this singular threshold, making it unfair to evaluate them over a full range of points. The flexibility of choosing a detection threshold is an additional benefit of using our event-based modeling approach, but will not be evaluated in this experiment.

Results for the detection of simulated events are shown in Figure 3. For 0.25 IoU (a more lenient setting), the event-based approach outperforms all epoch-based approaches. Additionally, one can see the impact of post-processing in an epoch-based setting. Both post-processing schemes improve upon an approach without post-processing.

For 0.75 IoU (a more strict setting), a similar pattern can be observed. The event-based approach outperforms the epoch-based approaches, and post-processing improves results for the epoch-based approaches. Post-processing based on morphological operations, a more intensive and task-specific scheme, further improves results in the sctricter setting.

C. Real-world data

Additionally, we show results on two real-world data sets, one containing EEG artefacts and the other one containing EEG with epileptic seizures.

a) EEG artefacts: The Temple University Artefact Corpus consists of various EEG artefact events, described in [32]. In the full data set, many multi-channel EEG recordings with channel-level annotations of artefact events are present. In this paper we focus on the muscle and chewing artefacts due to their large variability in duration. Both types are joined into a single artefact class. We train models on the

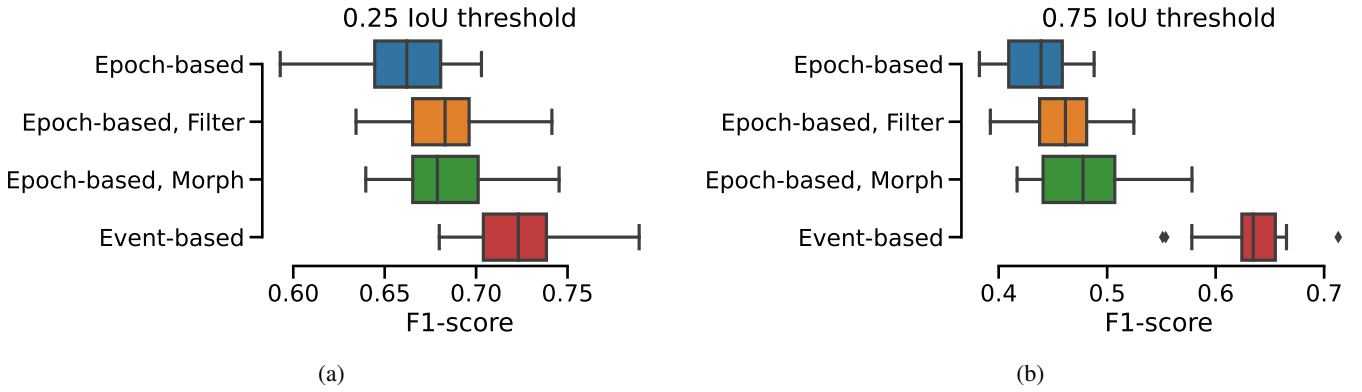


Fig. 3: Summary results of different training runs on simulated ECG artefacts. We evaluate performance using 0.25 and 0.75 IoU as the "threshold" to compute true/false positives and negatives to reflect a lenient and more strict evaluation setting. "Epoch-based" is an epoch-based approach *without* post-processing, "Epoch-based, Filter" is an epoch-based approach with median filtering, "Epoch-based, Morph" is an epoch-based approach combined with morphological operations, and "Event-based" is our event-based approach. Based on the Wilcoxon rank-sum test, our event-based approach achieves a significantly higher F1-score compared to all epoch-based approaches at the $p < 0.01$ significance level.

individual channels of this data set to predict the channel-level artefact annotations. The recordings are divided into training, validation, and test sets, making sure that recordings of the same individual are not split among the sets. Example artefact events can be found in Figure 4.

The event-based and epoch-based model both use the same backbone architecture. Post-processing for the epoch-based model is done with a median filter with filter length of $0.1\,\mathrm{s}$. Network details can be found in the Supplementary Material.

b) Epileptic seizures: The second real-world data set is the Temple University Seizure Corpus containing epileptic seizures [33]. The data set is made up of multi-channel EEG recordings, with epileptic seizures annotated at a general level (only indicating at which point in time a seizure occurs, not on which channel(s)). Example seizure events can be found in Figure 5

As comparison, we use the approach of [3], which won an international seizure detection challenge on this specific data set [34]. The approach consists of an epoch-based learning task with a U-Net architecture combined with extensive post-processing, tailored to seizure detection and the data set in question. For the sake of a fair comparison, we use the same backbone architecture as [3] in our event-based model, yet without the post-processing stage. Using the same backbone showcases the performance of our generic approach (without post-processing) compared to an epoch-based algorithm with heavily tailored post-processing. Network details can be found in the Supplementary Material.

1) Detection performance: For the real-world data sets, we use 0.5 IoU as a threshold as a meet-in-the-middle metric between an any-overlap scoring and a time-aligned event scoring [27]. Instead of reporting an aggregate measure like average precision, we compute precision at confidence thresholds corresponding to specific recall values (since, in a real-world setting, users would also need to decide on a confidence threshold). Next to the precision using 0.5 IoU, we also compute the proportion of predictions that still have positive overlap with a corresponding ground-truth event, but less than 0.5 IoU. This allows for a deeper understanding of what sort of predictions the two approaches produce.

Artefact and seizure detection precision is displayed in Figure 6 for different recall levels (computed using the IoU-based detection criterion, with 0.5 as threshold). Overlaps over 0.5 IoU are counted as *true positive* detections. Detected events having less than 0.5 IoU with a matched reference event, but which still have positive overlap, are indicated as <0.5 IoU. How well the approaches predict the duration of a detected event is gauged by the comparison between the true positive and <0.5 IoU detections. Predicted events that do not correspond to a ground-truth event are counted as *false positives*. A network that cannot detect events at a specific recall level is indicated as *No detection*.

For most recall levels, the event-based approach is outperforming the epoch-based one. Additionally, more events are found using the event-based approach compared to the epoch-based case, shown by the former's higher recall level. Note that both approaches do not reach 100 % recall. Unlike for binary classification, not all targets (events) get detected by moving the decision threshold to zero because of the IoU threshold to count predicted events as true positives.

The seemingly rising precision-recall curve for the epoch-based approach in the artefact data set is an unintuitive finding. Normal precision-recall curves are expected to show high precision at low recall, and low precision at high recall. This behavior is not shown in this case. The behavior can be explained by the decoding process of the epoch-based predictions. To predict events at specific recall levels, *confidence thresholds* need to be varied, which are set to correspond to a specific cutoff of the output to distinguish between *event* and *background*. Because of some noise in this output signal, two events can easily be detected as a single long event if the threshold is low. On the other side, a single event can easily be split into multiple shorter ones if the threshold is rather high. This sensitivity to a cutoff, and the fact that a 100 % recall

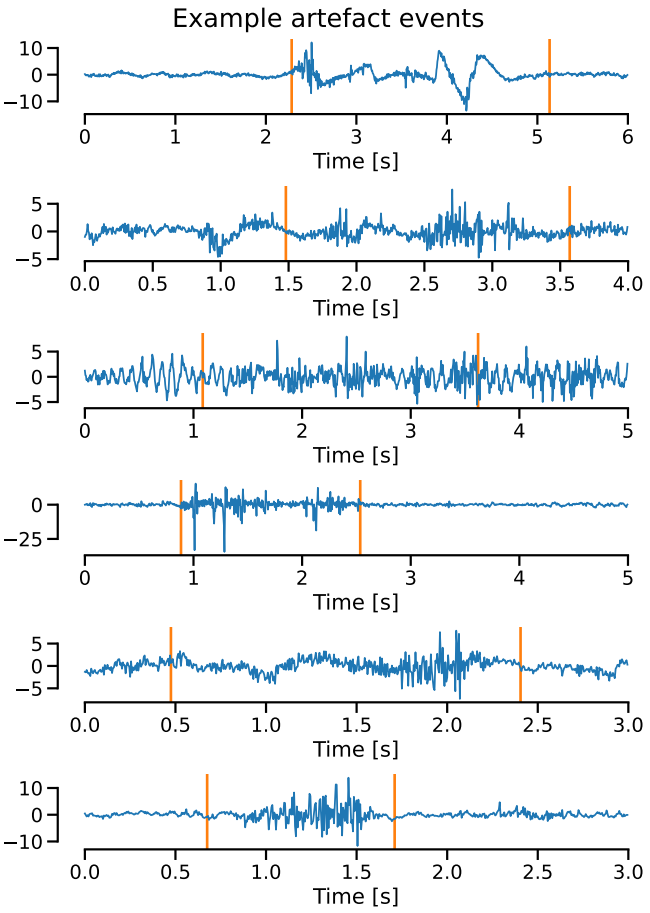


Fig. 4: Examples of EEG artefact events. The orange lines indicate the start and stop times of each event, as annotated in the Temple University artefact data set. Signal amplitude is unitless.

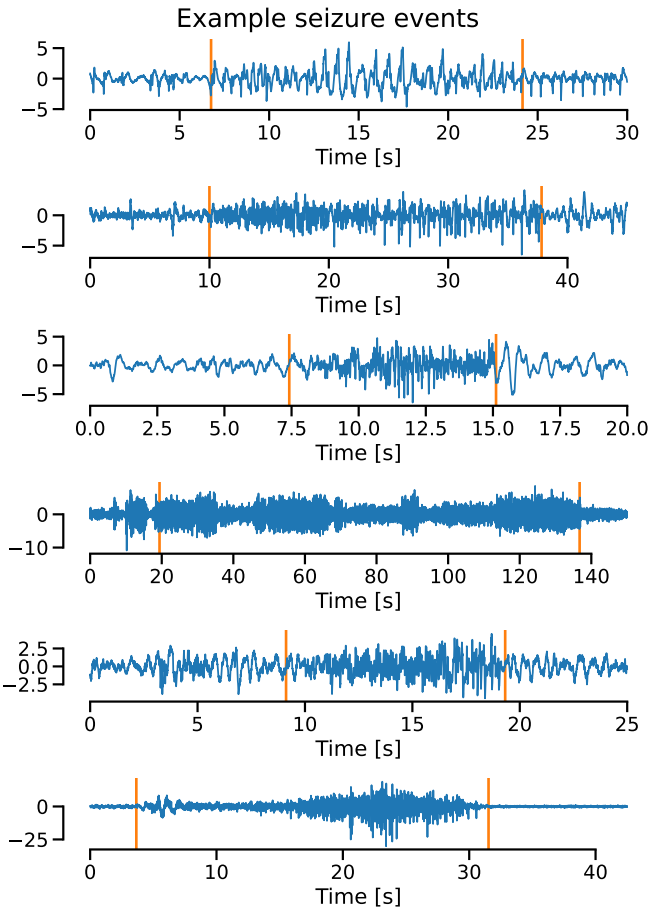


Fig. 5: Example of seizure events. These examples are single channels taken from multichannel EEG. Orange lines indicate the start and stop times of each event, as annotated in the Temple University seizure data set. Signal amplitude is unitless.

cannot be achieved, can result in an irregular precision-recall curve with a single Pareto-optimal operating point. Similar behavior is observed for the simulated events, as illustrated in the Supplementary Material.

2) Center and duration estimation: Evaluating performance solely based on IoU combines both "branches" of our event-based approach (center and duration predictions). In addition, we also evaluate the performance of both these aspects independently. To do so, predicted events that have positive overlap with their corresponding ground-truth events have their center point offsets and duration differences evaluated. In the case of the epoch-based approaches, the center point and duration are extracted directly from the events that are outputted by the post-processing stage, whereas for the event-based approach the center point and duration are directly obtained from the outputs of the neural network. We report *relative* errors for center point offsets and duration errors.

Regarding the center points and duration evaluation, both approaches are evaluated at the confidence threshold corresponding to the maximum common recall level (0.4 for the artefacts, and 0.33 for the seizures). Relative offsets between ground-truth and predicted centers are shown in Figure 7 for both data sets. For artefact detection, our event-based approach shows less variability around ground-truth center points compared to the epoch-based one. For seizure detection, the event-based approach shows less variability, with the median predicted center being closer to ground-truth.

For artefact detection center offsets, the event-based approach also shows no significant bias away from zero, while the epoch-based approach again shows a significant underestimation (setting the center earlier in time than the actual ground-truth center point), based on the t-test ($p < 0.01$). The event-based offsets are also significantly lower in variability, based on the Levene test ($p < 0.01$). For seizure detection center offsets, both approaches show no significant bias away from zero. The event-based offsets are significantly lower in variability than the epoch-based offsets ($p < 0.01$).

Relative duration prediction errors are shown in Figure 8 for both data sets. For artefact detection, the event-based median duration prediction is closer to the ground truth. Additionally, the event-based approach makes smaller errors in duration predictions (indicated by the lower interquartile range).

For artefact detection duration errors, the event-based approach shows no significant bias away from zero, while the epoch-based approach shows a significant overestimation, based on the t-test ($p < 0.01$). The event-based errors are

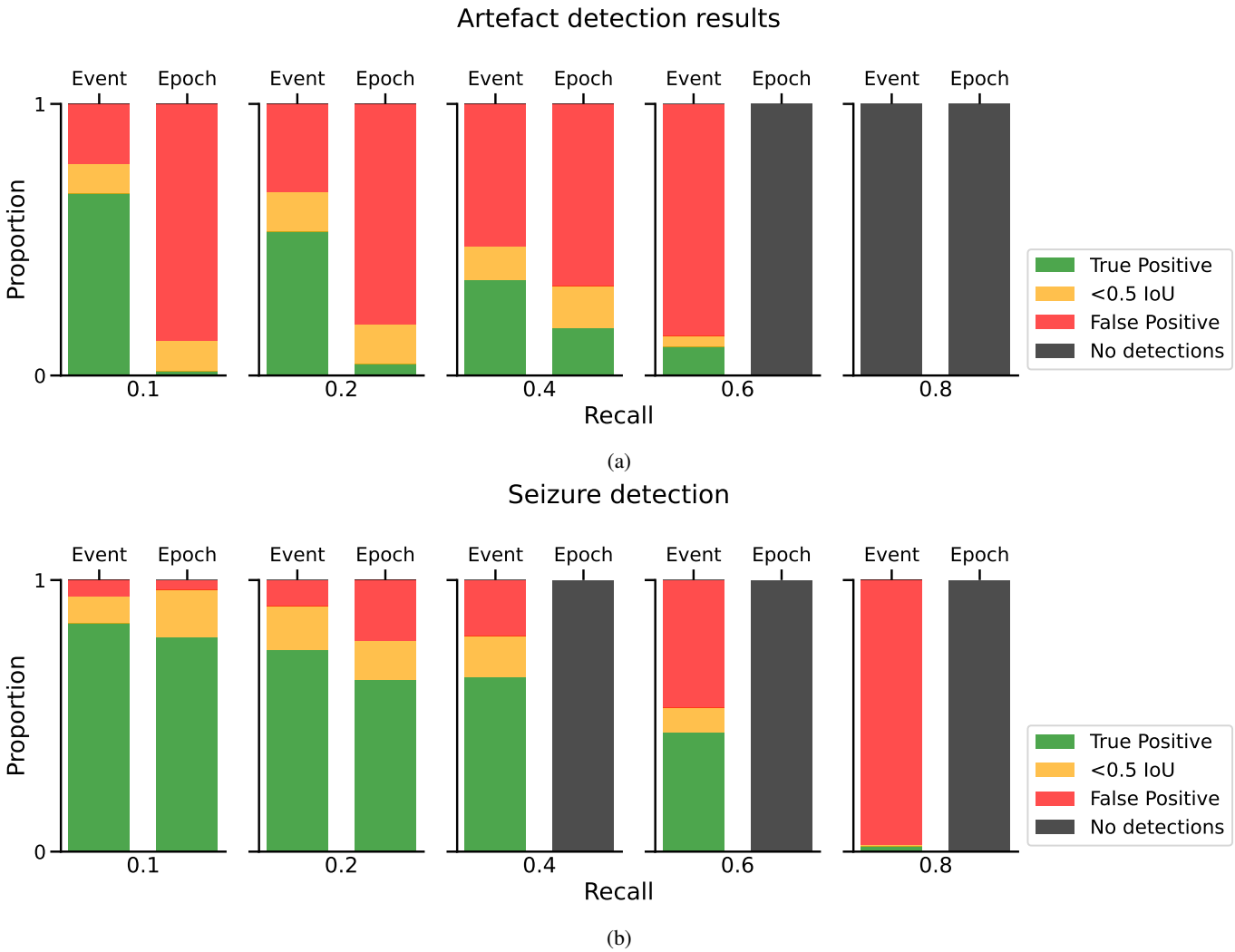


Fig. 6: Detection results for real-world EEG artefact and seizure events at different recall levels. *True positive* detections correspond to IoU ≥ 0.5. The proportion of predictions that have IoU < 0.5 but still have positive overlap are indicated in orange. *False positive* detections have no overlap with ground-truth events. *No detections* indicates that the algorithm cannot detect events at that recall level.

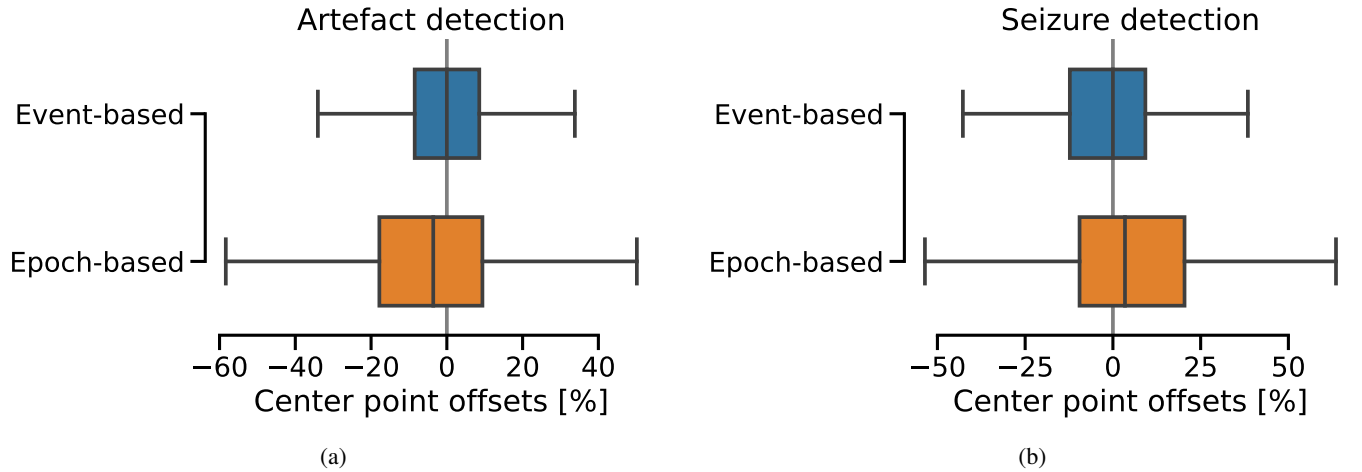
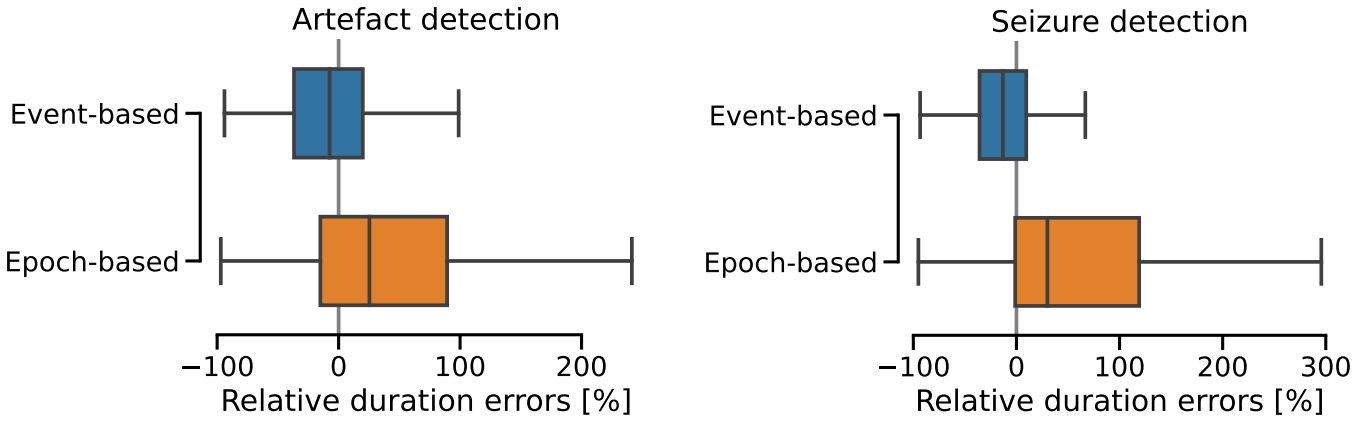


Fig. 7: Relative offsets between ground-truth and predicted centers (normalized by ground-truth duration). Positive values indicate that the predicted event center lies later in time than the ground truth. We consider all matched ground-truth and predicted events that show any overlap (green + orange class in Fig. 6)



(a) (b)

Fig. 8: Relative errors between ground-truth and predicted durations (normalized by ground-truth duration). We consider all matched ground-truth and predicted events that show any overlap (green + orange class in Fig. 6)

also significantly lower in variability, based on the Levene test ($p < 0.01$). For seizure detection duration errors, both approaches show a significant bias away from zero ($p < 0.01$). The event-based errors are significantly lower in variability than the epoch-based errors ($p < 0.01$).

3) Impact of training set size: To better understand the impact of training set size on event-based and epoch-based approaches, we investigate performance "growth" with growing training set size. For varying sizes of both real-world data sets, the two approaches are trained and later evaluated on the respective test sets. To investigate the relation of performance with training set size, a scalar performance measure is required. To this end, the F1-score of both models is computed at the optimal confidence threshold point (corresponding to the maximum F1-score on the validation set for each training run), and plotted as a function of training set size. For every training run, we train the models with the same amount of batches as training on the full training set would take (the number of epochs is corrected for the smaller training set sizes).

Performance growth for both data sets is shown in Figure 9, together with a logistic growth model fitted to the scatter points. Crucially, fitting this growth model is done for illustrative purposes only (to get a feeling for the asymptotic behavior of the two methods). It is not meant as a confident extrapolation (predicting the networks' performance for specific training set sizes). For artefact detection, asymptotic F1-scores are estimated for the event-based approach at 0.35, and at 0.13 for the epoch-based approach by the logistic growth model. For seizure detection, asymptotic F1-scores are estimated for the event-based approach at 0.42, and at 0.31 for the epoch-based approach by the growth model. Visually, it can be concluded that the EEG artefact data seems to contain enough events to properly train an event-based network. Both the scatter points and the growth model show asymptotic behavior. The epoch-based approach, on the other hand, struggles for artefact events. We believe these low F1-scores reflect the epoch-based approach's high sensitivity to a decision threshold (as shown by the singular operating points in Figure 6a and the ECG results in the Supplementary Material). For seizure detection, the training set is large enough to train an epoch-based approach. The same asymptotic behavior is not shown for the event-based approach. It seems that the event-based approach would benefit from more training examples. The actual performance measurements, however, are rather scattered, making it difficult to draw hard conclusions for the seizure detection task.

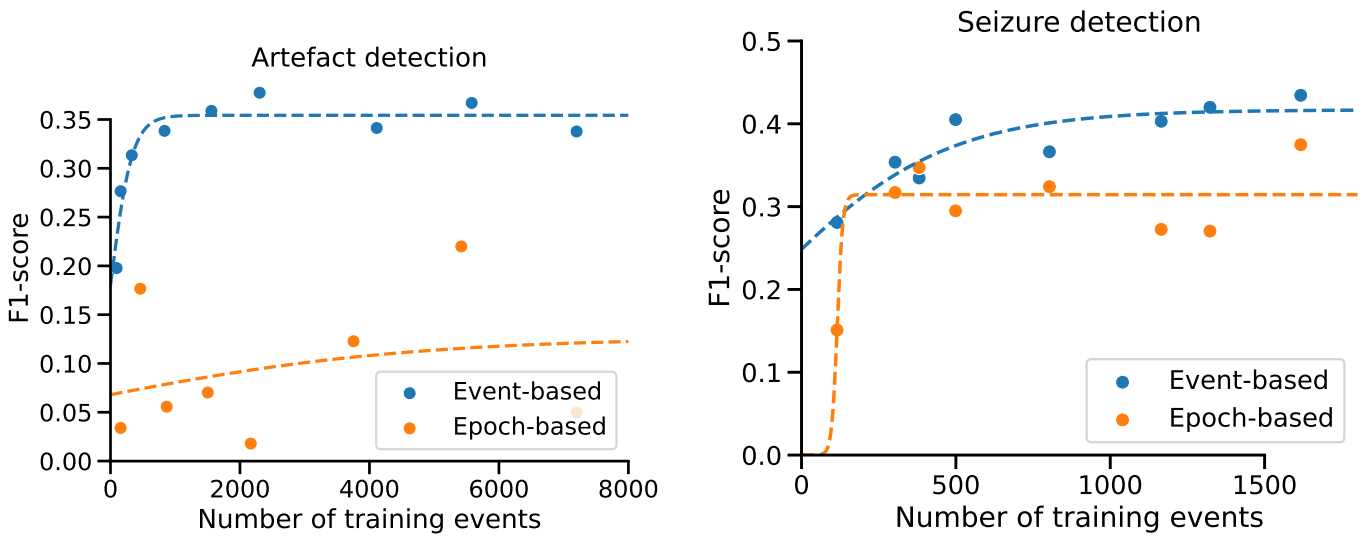
IV. Discussion

A. Discussion of experimental results

Using the simulated event data set, we show the power of event-based modeling. The event-based approach clearly outperforms the epoch-based approaches. We want to emphasize that we do not claim superior performance in general, only strictly with these approaches. Our main goal is to show the ease with which a performant model can be designed with event-based modeling compared to epoch-based modeling.

On real-world data, the event-based approach outperforms the epoch-based benchmarks. Looking at the EEG artefact results, the event-based approach shows improved precision at all recall levels (Note that this is a very challenging data set). For the artefacts, additionally, our event-based approach shows improved duration prediction, based on the higher proportion of "True positives" relative to "<0.5IoU" in Figure 6a and in the prediction error evaluation of Figure 8a.

Regarding EEG seizures, the event-based approach generally outperforms the epoch-based benchmark. Only for 0.1 recall, the epoch-based benchmark manages to find slightly more events (as evidenced by the combination of the green and orange part of Figure 6b). At the same time, the event-based predictions are of slightly higher quality (as evidenced by the higher proportion of "True positives", i.e., higher proportion of >0.5 IoU overlap). Note that this epoch-based benchmark recently scored top place in a competition on this exact data set, and is highly tuned. Our event-based approach uses the base model of the benchmark and turned it into an event-based model by the addition of a center and duration "head".



(a) (b)

Fig. 9: Performance growth for growing training set size. Scatter points represent the measured values for specific numbers of training events. Dashed lines show a logistic growth model fitted to the scatter points.

Despite the potential benefit of performance due to an event-based approach, we want to emphasize that the largest benefit of our framework lies in removing the need for post-processing in epoch-based approaches. This post-processing was different for the three data sets, and will be different again for new data sets.

B. Using an event-based framework

We have proposed an event-based deep learning framework for time series, and have applied it to simulated ECG artefacts, and real-world artefact and seizure detection in EEG. Due to the end-to-end nature of our approach, and the neural network backbone learning its own data representation, our framework is more broadly applicable than these use cases.

The major difference, and benefit, of event-based modeling compared to classical epoch-based approaches, is the lack of case-specific post-processing. Using event-based modeling, one can go directly from model output to events, whereas epoch-based approaches will always need some kind of tailored translation step before events can be listed. End-to-end event-based models drop the requirement for domain-specific post-processing rules. These can involve the expected event duration, duration ranges, how quickly events can follow each other, etc. While event-based models might also benefit from such post-processing, it is not a crucial step as opposed to existing approaches. Our framework can automatically learn most of those patterns from data, as is evident from our experiments. It should be noted that we do not claim that an event-based approach will always learn all patterns and will always outperform epoch-based approaches with well-designed post-processing. When extensive domain expertise is available, epoch-based approaches relying on features and post-processing rules inspired by this expertise can potentially outperform a generic event-based model, especially when limited training data is available.

One of the major benefits of working within an event-based framework, compared to segmenting signals into epochs, is the more intuitive nature of labels. The way our framework uses event labels matches closely to how human annotators would work with these labels (defining a start and stop time, which is equivalent to a center point and duration). This close match allows for easier feedback from human experts when developing a machine learning solution. This should be compared to the classical epoch-based approach, where labeled events need to be translated into a sequence of classification targets (with potential ambiguity at event borders), and back to events at inference time. This can add substantial friction to the development process, and hinder expert feedback.

Our current framework is conceptually simple: learn to predict a center point, and learn to predict a duration, which is expected to be symmetric around this center point. These two tasks are performed independently (albeit based on a common feature map), and the two loss terms are computed and optimized independently. If, at test time, the center point prediction is off by some margin, the training process does not guarantee meaningful duration predictions at this "offset" center point. However, it is expected that small offsets would still lead to meaningful duration predictions, given that the boundaries of the events are quite ambiguous in the first place (i.e., also during training). In our experiments, we indeed observed that the duration predictions around the (ground-truth) center point positions were still close to the actual event duration. However, our framework can be extended if robustness issues are expected in a novel application. An interesting starting point in that regard are the "asymmetric" size predictions of [23] in the context of computer vision, where the model is explicitly trained to also predict a meaningful bounding box at different points than the ground-truth center point, at the cost of a more complex loss function and increased compute time. In our time-series context, this would mean that for each time

point within an event, we would predict 2 duration values: one to the left and one to the right of the current point (indicating the distance to the start and end of the event). It is noted that we have also implemented such an asymmetric duration prediction, but it did not lead to significant improvements on the aforementioned data sets.

Applying our event-based framework in real-life applications requires setting a "threshold strategy" when decoding events. For our experimental results, we attempt to show the full range of outcomes (Precision-recall curves for simulated data in the Supplementary Material, and Figs. 6a and 6b). Depending on the use case, different settings are relevant (higher precision, tolerating low recall, and high recall, tolerating lower precision). It should be noted that the decoding schemes allows for a wide range of "threshold strategies" outside of the traditional approach of fixing a threshold beforehand. If one expects no more than N events in a recording based on expert information, one can select the top N predictions, regardless of actual model confidence. In another setting, one can devise a scheme that dynamically searches for the decoding "noise floor", i.e., what confidence values are associated to a large number of probably-spurious detections and set a confidence threshold above this dynamic noise floor.

Some event detection applications might require real-time detection of the events. Herein lies another difference in using an event-based or epoch-based approach. By design, our event-based framework will require that (most of) the event of interest has been observed. Then, because the model is encouraged to model the full characteristics of an event, the model might require more context after an event has occurred to properly detect it. In contrast, an epoch-based approach can be designed to detect an event occurrence before it has been fully observed (by choosing epoch duration and post-processing schemes in line with real-time limitations.) Real-time detection can cover multiple use cases. For use cases where the desired outcome is detecting an event directly after it has been observed, one can potentially use our event-based framework. For use cases where the desired outcome is a detection as soon as possible, before the event has concluded, epoch-based approaches might be more relevant.

A potential limitation or difficulty in using an event-based framework is the need for training data. Relevant features and event characteristics are learned jointly, relying on enough training examples to do so. A key aspect about these training examples is the diversity of durations. Our framework learns to directly predict durations, so it requires a broad range of example durations to learn from. The real-world artefact and seizure data sets both cover a wide range of durations, but are heavily skewed towards shorter events. The impact of duration distribution on performance is unknown at present. One can imagine, for example, that shorter durations are easier to predict if shorter events are more consistent in nature than longer events but this remains to be investigated. Aside from the nature of example events, one should also consider the amount of examples. As seen in the seizure detection performance growth, there might be potential improvements with more examples. The epoch-based approach, on the other hand, shows "saturated" performance, with no clear indication that it might benefit from more examples. Related to the amount of examples is also the training time. Intuitively, one can reason that learning a meaningful representation to predict an event's center and duration is more difficult than deciding whether a particular epoch belongs to the "event" class of "background" class. Our framework requires enough training examples to do so, but also enough training time to learn the relevant patterns.

V. Conclusion

In this paper, we have proposed and showcased an event-based approach to a broad class of event detection problems in biomedical signal processing. The model can directly detect events of variable duration in long signal recordings. In contrast to existing epoch-based methods, we require no post-processing scheme to translate predictions into a set of events. Our model can be extended to other biomedical event-detection tasks and to other signal processing tasks where signal events are involved.

References

[1] J. Zhang *et al.*, "Automatic annotation correction for wearable EEG based epileptic seizure detection," *Journal of Neural Engineering*, vol. 19, no. 1, p. 016 038, 2022.

[2] K. Vandecasteele *et al.*, "Visual seizure annotation and automated seizure detection using behind-the-ear electroencephalographic channels," *Epilepsia*, vol. 61, no. 4, pp. 766–775, 2020.

[3] C. Chatzichristos *et al.*, "Epileptic seizure detection in EEG via fusion of multi-view attention-gated U-Net deep neural networks," in *2020 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, 2020, pp. 1–7. DOI: 10.1109/SPMB50085.2020.9353630.

[4] A. H. Ansari *et al.*, "Neonatal seizure detection using deep convolutional neural networks," *International journal of neural systems*, vol. 29, no. 04, p. 1 850 011, 2019.

[5] H. Li and Y. Guan, "DeepSleep convolutional neural network allows accurate and fast detection of sleep arousal," *Communications biology*, vol. 4, no. 1, pp. 1–11, 2021.

[6] P. M. Kulkarni *et al.*, "A deep learning approach for real-time detection of sleep spindles," *Journal of neural engineering*, vol. 16, no. 3, p. 036 004, 2019.

[7] A. N. Olesen *et al.*, "MSED: A multi-modal sleep event detection model for clinical sleep analysis," *arXiv preprint arXiv:2101.02530*, 2021.

[8] S. Chambon *et al.*, "DOSED: A deep learning approach to detect multiple sleep micro-events in EEG signal," *Journal of neuroscience methods*, vol. 321, pp. 64–78, 2019.

[9] M. Yeo *et al.*, "Respiratory event detection during sleep using electrocardiogram and respiratory related signals: Using polysomnogram and patch-type wearable device data," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 2, pp. 550–560, 2021.

[10] E. Urtnasan *et al.*, "Automatic detection of sleep-disordered breathing events using recurrent neural networks from an electrocardiogram signal," *Neural computing and applications*, vol. 32, pp. 4733–4742, 2020.

[11] C. Varon *et al.*, "A novel algorithm for the automatic detection of sleep apnea from single-lead ecg," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 9, pp. 2269–2278, 2015.

[12] U. Erdenebayar *et al.*, "Deep learning approaches for automatic detection of sleep apnea events from an electrocardiogram," *Computer methods and programs in biomedicine*, vol. 180, p. 105 001, 2019.

[13] H. Phan *et al.*, "Seqsleepnet: End-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 3, pp. 400–410, 2019.

[14] J. Moeyersons *et al.*, "Artefact detection and quality assessment of ambulatory ecg signals," *Computer methods and programs in biomedicine*, vol. 182, p. 105 050, 2019.

[15] J. Moeyersons *et al.*, "Artefact detection in impedance pneumography signals: A machine learning approach," *Sensors*, vol. 21, no. 8, p. 2613, 2021.

[16] J. Behar *et al.*, "Ecg signal quality during arrhythmia and its application to false alarm reduction," *IEEE transactions on biomedical engineering*, vol. 60, no. 6, pp. 1660–1666, 2013.
[17] L. Zhao *et al.*, "Quantitative signal quality assessment for large-scale continuous scalp eeg from a big data perspective," *Physiological Measurement*, 2022.
[18] A. Malafeev *et al.*, "Automatic artefact detection in single-channel sleep EEG recordings," *Journal of sleep research*, vol. 28, no. 2, e12679, 2019.
[19] L. Webb *et al.*, "Automated detection of artefacts in neonatal EEG with residual neural networks," *Computer Methods and Programs in Biomedicine*, vol. 208, p. 106 194, 2021.
[20] A. Craik *et al.*, "Deep learning for electroencephalogram (eeg) classification tasks: A review," *Journal of neural engineering*, vol. 16, no. 3, p. 031 001, 2019.
[21] X. Zhou *et al.*, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.
[22] P. Virtanen *et al.*, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: 10.1038/s41592-019-0686-2.
[23] Z. Liu *et al.*, "Training-time-friendly network for real-time object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 11 685–11 692.
[24] T.-Y. Lin *et al.*, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.
[25] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 734–750.
[26] J. Redmon *et al.*, "You Only Look Once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016.
[27] S. Ziyabari *et al.*, "Objective evaluation metrics for automatic classification of EEG events," *arXiv preprint arXiv:1712.10107*, 2017.
[28] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *European conference on computer vision*, Springer, 2014, pp. 740–755.
[29] G. D. Clifford *et al.*, "Af classification from a short single lead ecg recording: The physionet/computing in cardiology challenge 2017," in *2017 Computing in Cardiology (CinC)*, IEEE, 2017, pp. 1–4.
[30] G. B. Moody *et al.*, "A noise stress test for arrhythmia detectors," *Computers in cardiology*, vol. 11, no. 3, pp. 381–384, 1984.
[31] R. M. Haralick *et al.*, "Image analysis using mathematical morphology," *IEEE transactions on pattern analysis and machine intelligence*, no. 4, pp. 532–550, 1987.
[32] A. Hamid *et al.*, "The Temple University artifact corpus: An annotated corpus of EEG artifacts," in *2020 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, IEEE, 2020, pp. 1–4.
[33] V. Shah *et al.*, "The Temple University hospital seizure detection corpus," *Frontiers in neuroinformatics*, vol. 12, p. 83, 2018.
[34] N. Neurotech and NeuroTechX. "Neureka^{TM} 2020 epilepsy challenge." (), [Online]. Available: https://neureka-challenge.com/ (visited on 06/12/2020).