KU LEUVEN

STADIUS
Center for Dynamical Systems, Signal Processing and Data Analytics

Citation/Reference	Seeuws, Nick, Maarten De Vos, and Alexander Bertrand. **"A Human-in-the-Loop Method for Annotation of Events in Biomedical Signals."** *IEEE Journal of Biomedical and Health Informatics* (2024).
Archived version	Author manuscript: the content is identical to the content of the published paper, but without the final typesetting by the publisher
Published version	https://doi.org/10.1109/JBHI.2024.3460533
Journal homepage	https://ieeexplore.ieee.org/abstract/document/10679933
Author contact	nick.seeuws@esat.kuleuven.be
Abstract	Objective: Building large-scale data bases of biomedical signal recordings for training artificial-intelligence systems involves substantial human effort in data processing and annotation. In the case of event detection, experts need to exhaustively scroll through the recordings and highlight events of interest. Methods: We propose an iterative annotation support algorithm with a human in the loop to improve the efficiency of the annotation process. Our algorithm generates proposal events based on an event detection model trained on incomplete annotations. The human only needs to verify candidate events proposed by the tool instead of scrolling through the entire data set. Our algorithm iterates between proposal generation and verification to leverage the human-in-the-loop feedback to obtain a growing set of event annotations. Results: Our algorithm finds a substantial amount of events at a fraction of the human time spent when comparing with a benchmark method and the normal manual process, finding all events in one data set and 70\% of events in another with the human-in-the-loop only viewing 20\% of the data. Conclusion: Our results show that combining human and computer effort can substantially speed up the annotation process for events in biomedical signal processing. Significance: Due to its simplicity and minimal reliance on task-specific information, our algorithm is broadly applicable, unlocking substantial improvements in the scalability and efficiency of biomedical signal annotation.

(article begins on next page)

A Human-in-the-Loop Method for Annotation of Events in Biomedical Signals

Nick Seeuws, Maarten De Vos, and Alexander Bertrand

***Abstract*— *Objective:* Building large-scale data bases of biomedical signal recordings for training artificial-intelligence systems involves substantial human effort in data processing and annotation. In the case of event detection, experts need to exhaustively scroll through the recordings and highlight events of interest. *Methods:* We propose an iterative annotation support algorithm with a human in the loop to improve the efficiency of the annotation process. Our algorithm generates proposal events based on an event detection model trained on incomplete annotations. The human only needs to verify candidate events proposed by the tool instead of scrolling through the entire data set. Our algorithm iterates between proposal generation and verification to leverage the human-in-the-loop feedback to obtain a growing set of event annotations. *Results:* Our algorithm finds a substantial amount of events at a fraction of the human time spent when comparing with a benchmark method and the normal manual process, finding all events in one data set and 70% of events in another with the human-in-the-loop only viewing 20% of the data. *Conclusion:* Our results show that combining human and computer effort can substantially speed up the annotation process for events in biomedical signal processing. *Significance:* Due to its simplicity and minimal reliance on task-specific information, our algorithm is broadly applicable, unlocking substantial improvements in the scalability and efficiency of biomedical signal annotation.**

***Index Terms*— Biomedical Signal Processing, Deep Learning, Neural Network, Human-Computer Interaction**

I. Introduction

A. Problem statement

Large-scale data sets underpin many contemporary works in biomedical data processing [1]–[3]. Massive data sets are processed using machine learning to automatically find patterns of interest. Traditionally, the field relied on statistical models and hand-crafted features [4]–[10]. Deep learning models are increasingly used to automatically learn task-specific features in an end-to-end manner [11]–[16], leveraging large data sets to train larger and better models. Building large data sets for biomedical data processing, and push the state of the art further, requires a massive effort from human experts to annotate data. These experts annotate training sets to develop machine learning algorithms. To scale the range and size of current data sets, improving the efficiency of the data annotation process is crucial, allowing experts to process more data with less effort. This paper looks at the problem of making the data annotation process more efficient for the case of event detection in biomedical signals.

Events in biomedical signal processing are specific patterns of interest appearing in the background signal at specific points in time. The scope of events is broad, and difficult to describe a priori. Examples include epileptic seizures [4], [5], [13], [14], sleep events[11], [12], [17], and signal artefacts [6], [7], [18]–[22]. The annotation process for these tasks produces a list of (t_{start}, t_{stop}) tuples corresponding to the start and stop time of events. This form is different from classical machine learning data, which come in the form of (data point, target) tuples, making direct application of existing machine learning solutions difficult [16].

The low prevalence of events in biomedical signal recordings makes annotation challenging. For example, in a recent study testing seizure detection in wearables [23], the authors found, on average, 1 seizure event per 3 days of signal recordings. Manual exhaustive annotation requires experts to scroll through all signal recordings, interpreting the signal's behavior before and after potential patterns of interest. This process requires years of experience and is time-consuming and expensive, stressing the need for efficiency.

Improving efficiency of the annotation process is challenging due to a chicken-and-egg problem: machine learning solutions for annotation support require annotations to be built, but the lack of annotations is the reason for designing such algorithms. A well-performing annotation support algorithm needs to work in a low-annotation regime, while at the same time also respecting the full breadth and depth of event characteristics present in a data set.

We propose a generic iterative annotation support algorithm for events in biomedical signals with a human in the loop. We consider an event to be fully specified by a (start time, stop time) tuple, and leave the specification of the "pattern of interest" to an end user, adding to the genericity of our algorithm. The algorithm generates proposal events based on

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). This research work was carried out at the ESAT Laboratory of KU Leuven, in the frame of the FWO Research Project: 'Deep, personalized epileptic seizure detection', G0D8321N.

N. Seeuws, M. De Vos, and A. Bertrand 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

N. Seeuws, M. De Vos, and A. Bertrand are affiliated to Leuven.AI - KU Leuven Institute for AI, B-3000, Leuven, Belgium.

an event detection model trained on incomplete annotations. The human-in-the-loop verifies candidate events proposed by the algorithm instead of scrolling through the entire data set. Our algorithm is conceptually simple and does not rely on domain-specific knowledge.

B. Related work

Annotation support algorithms commonly use an iterative propose-verify process with a human in the loop [24]–[32]. These algorithms can be broadly categorized by: a) the object of interest, b) proposal generation, and c) the verification process. Most existing algorithms focus on classifying single data points. In contrast, our approach aims to localize low-prevalence events within biomedical signal backgrounds, a fundamentally different and more complex problem than mere classification [16], [25], [26].

The prior works of [25], [26] involve predefined segmentation and feature extraction steps to localize events, which typically require domain-specific adaptations. In general, existing methods involve complex, task-specific techniques to generate proposals [24], [26], [27], limiting their adaptability. Verification processes also vary. Some require human experts to verify every annotation to ensure reliability [24], [25], while others only review a subset, potentially leading to lower annotation quality [26], [27].

Our method contrasts itself by leveraging a simpler, more generic proposal generation process that does not rely on predefined segmentation or extensive feature extraction, enhancing flexibility and reducing the need for domain-specific knowledge. This simplicity allows for a broader application across various biomedical contexts. Additionally, our verification process ensures all proposed annotations are reviewed by a human experts, combining the rigor of full verification with the efficiency of algorithmic pre-selection.

Our algorithm leverages key principles from machine learning and human-computer interaction to achieve efficient event annotation. The proposal generation mechanism draws inspiration from uncertainty sampling in active learning [33], directing human effort towards the most informative samples. The iterative nature of our approach parallels expectation-maximization algorithms [34], gradually refining annotations and model performance. Finally, our method reflects the idea of incremental learning [35], continuously updating its knowledge base with newly verified annotations.

Problems in the field of active learning are closely related to interactive annotation[33]. While active learning seeks to minimize human effort in training models, our annotation support algorithm focuses on exhaustively annotating the dataset at hand with minimal effort, instead of just optimizing model training. This focus on data coverage is crucial for constructing high-quality, reliable biomedical event annotations and differentiates our work from typical active learning settings. However, active learning strategies can be integrated in annotation support algorithms [24], [27].

C. Summary and contributions

We introduce a generic, easy-to-implement annotation support algorithm for event annotation in biomedical signals, using an iterative propose-verify loop driven by deep learning, illustrated in Figure 1. Our approach does not rely on task-specific features, segmentation schemes, or post-processing. It draws inspiration from "pseudo-labels" in semi-supervised learning to generate proposal annotations for human verification. Our approach improves the efficiency of the annotation process and, at the same time, ensures trustworthiness of the results.

To summarize, our contributions are as follows:

- We introduce a scalable and flexible algorithm that is particularly effective in scenarios with low event prevalence,
- By leveraging a deep learning backbone, our algorithm does not rely on domain-specific features (as opposed to [25], [26]), making it broadly applicable across various biomedical signal processing tasks,
- The simplicity of our algorithm's design allows for easy implementation and promotes adaptability, enabling researchers to deploy it in novel settings without extensive modification.

Section II explains the algorithm. Section III covers our experiments and results. Section IV discusses the lessons learned from these experiments and broader aspects of the algorithm. Section V concludes the paper.

II. Methods

Our annotation support algorithm involves iterative steps of computer-based proposal generation and human-centric proposal verification. The proposal generation step is split up into a learning phase and the actual proposal generation phase. Algorithm 1 shows a high-level summary in pseudocode.

The algorithm starts with a small set of annotations previously obtained by a human ("positive" event annotations $\mathcal{A}_{start}$). These can be obtained through browsing signal recordings or exhaustively annotating a few recordings. How much human effort should be invested prior to the iterative process is a trade-off between how difficult one expects "blind annotations" to be, and how difficult confirming and correcting proposals is expected to be. As our experiments show, the approach can get started with very few annotations, but may require more iterations in such cases. We discuss the learning, proposal generation, and human verification steps in detail below.

A. Learning from incomplete annotations

This step learns event patterns from the current (potentially small) set of annotations $\mathcal{A}$. Events are short "bursts" of a pattern of interest standing out from the background signal, with large variability in duration. Traditionally, detection algorithms require intricate, task-specific design decisions [1]. Our annotation support algorithm relies on an underlying machine learning model for event detection. Ideally, the underlying model is a generic one, making the annotation support algorithm broadly applicable.

We use the work of [16] as the underlying event detection model, although in principle any machine learning algorithm can be used. Nevertheless, the underlying event detection model has to be able to learn in the presence of many

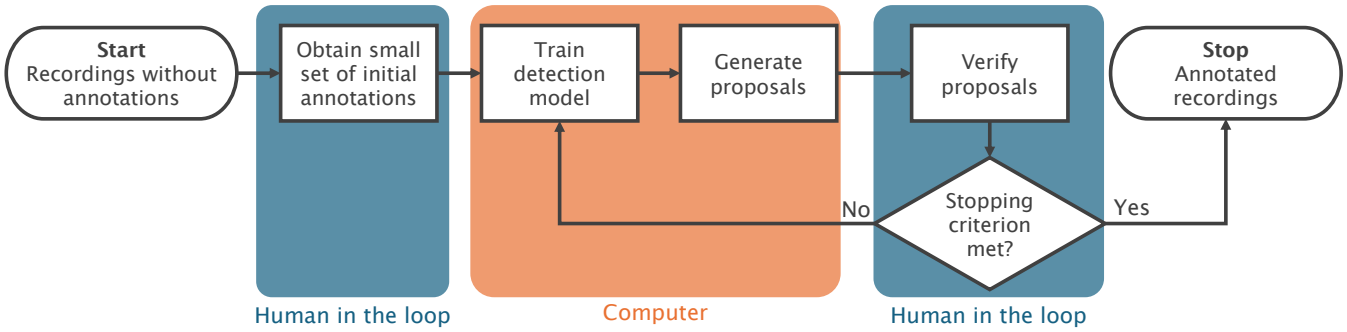


Fig. 1: Flowchart of our annotation support algorithm. The process begins with a small set of initial annotations and iterates between computer-driven steps (model training and proposal generation) and human-in-the-loop verification. This cycle continues until a pre-defined stopping criterion is met, resulting in an annotated dataset.

Algorithm 1: High-level summary of our annotation support algorithm in pseudocode

Data: Data set of signal recordings $\mathcal{D}$, initial set of event annotations $\mathcal{A}_{start}$, initial set of background (negative) annotations $\mathcal{B}$ (note that $\mathcal{B}$ can be empty)
Result: Full set of annotations $\mathcal{A}$
$\mathcal{A} \leftarrow \mathcal{A}_{start}$;
while *not* `MetStoppingCriterion()` **do**
  model $\leftarrow$ `LearnFromAnnotations`$(\mathcal{A},\mathcal{D})$;
  $\mathcal{A}_{proposal} \leftarrow$ `GenerateProposals`(model,$\mathcal{A}$, $\mathcal{D}$, $\mathcal{B}$) `/* Alg. GenerateProposals for details */` ;
  $\mathcal{A}_{verified} \leftarrow$ `VerifyProposals`$(\mathcal{A}_{proposal})$;
  $\mathcal{A} \leftarrow \mathcal{A} \cup \mathcal{A}_{verified}$;
  $\mathcal{B} \leftarrow \mathcal{B} \cup (\mathcal{A}_{proposal} \setminus \mathcal{A}_{verified})$;
end

Function GenerateProposals

Data: Event detection model model, set of existing event annotations $\mathcal{A}$, data set of signal recordings $\mathcal{D}$, set of background (negative) annotations $\mathcal{B}$
Result: Set of proposal annotations $\mathcal{A}_{proposal}$
$\mathcal{A}_{proposal} \leftarrow \emptyset$;
$\mathcal{P} \leftarrow$ `GeneratePredictions`$(model, \mathcal{D})$;
while *count*$(\mathcal{A}_{proposal})$ < N **do**
  a $\leftarrow$ `FindMostConfident`$(\mathcal{P})$;
  if a $\notin \mathcal{A} \wedge$ a $\notin \mathcal{B}$ **then**
    $\mathcal{A}_{proposal} \leftarrow \mathcal{A}_{proposal} \cup$ a;
  end
  $\mathcal{P} \leftarrow \mathcal{P} \setminus$ a;
end

false negatives, since false negatives dominate true positives in the first iterations of the algorithm (many events present in the signal are not yet annotated, and thus considered as "background" for the learner). The model of [16] can train in such a setting. It is an end-to-end deep learning model for the detection of signal events which does not require the splitting of the data in input segments of equal length, thereby avoiding post-processing to find event boundaries. Crucially, the model relies on deep learning, avoiding the crafting of task-specific features. The model takes a signal as input and, using a sequence-to-sequence architecture, predicts a center point and duration "signal". Peaks in the predicted center signal represent an event identified by the model, and the value of the duration signal at the corresponding time point represents this event's duration. The model is encouraged to fully capture characteristics of target events by jointly learning events' center points and durations. For the experiments explained in Section III, the specific backbone architectures are discussed in Appendix B.

We train at every iteration with the working set of annotations $\mathcal{A}$ using the full data set (i.e., also including all non-annotated events), aiming to learn event patterns while having the model recognize actual background but struggle unannotated events (and are thus initially considered as background for the sake of training a model). After training, we exploit the model's potential confusion on un-annotated events to generate proposals (note that we do not rely on annotated background segments, we only require $\mathcal{A}$ for training the model). To limit reinforcement of bias, we start the learning step from a randomly initialized network at every iteration.

B. Proposal generation

Proposal generation is based on the idea of pseudo-labels from semi-supervised learning [36]. Pseudo-labels are high-confidence predictions on unlabeled data based on a model trained on labeled data. We locate high-confidence event predictions in the data that are *not annotated* (i.e., not in $\mathcal{A}$ and not in $\mathcal{B}$, the confirmed background segments), relying on "confusion" during model training regarding the currently un-annotated events. To measure model confidence in an event prediction, we take the value of the center point signal corresponding to a predicted event[16]. The N highest-confidence event predictions that are not in $\mathcal{A}$ or $\mathcal{B}$ become the set of event proposals. Setting N is an important hyper-parameter of our approach which introduces a trade-off between the annotation

budget per iteration and the amount of new information that can be leveraged from iteration to iteration. We elaborate on this trade-off in an ablation study in Section III-E.

We track confirmed background segments $\mathcal{B}$ to avoid proposing the same background segment in multiple iterations. In the first iteration, $\mathcal{B}$ is either empty or contains confirmed background examples obtained when constructing $\mathcal{A}_{start}$.

C. Human verification

For each event, the human verifies whether the proposal corresponds to an actual, desired event. For confirmed events the human can adjust the proposed start and stop times to better correspond to the use case or correct small errors. If the proposal is wrong, the proposal is added to $\mathcal{B}$.

Verified events are added to the set of event annotations $\mathcal{A}$. Next, the process restarts with this updated set of annotations, or stops because of a stopping criterion. In practice, the stopping criterion will likely be tied to a "human time budget" (the human annotator can only verify X proposals in total), or tied to the "true positive rate" of proposal events (if 99% of proposals are consistently wrong a human annotator will stop annotating).

Our experiments simulate the human-in-the-loop by relying on ground-truth annotations for the verification step. Because of this simulation, our stopping criterion is determined by algorithm runtime. For the ECG data set we run the algorithm for 20 iterations. For the EEG data set we run the algorithm for 12 iterations (reflecting the substantial increase in data set size and training time for the underlying event detection model). Data sets are discussed in detail below in Section III-B.

III. Experiments and Results

A. Evaluating performance

The primary measure of performance is the human effort required to annotate a certain number of events or an entire dataset[24], [25], [37]. Direct measurement of this effort is challenging and costly, as it requires experts with substantial domain-specific knowledge and can be influenced by individual expertise levels. To ensure a consistent, objective measure of human effort, we use two proxy metrics: the total duration of data segments viewed by a simulated human-in-the-loop, and the number of decisions the (simulated) annotator makes. These proxies reflect the primary dimensions of the human effort involved in event annotation: visual examination of signal segments and decision-making. The actual human effort is expected to be strongly related to the aggregated duration of reviewed segments and the total amount of proposals, although time taken to process each segment may vary based on segment content, and some proposals may be easier to verify than others. Using such proxies is common in the field of interactive labeling [37], providing a reliable estimate of the effort reduction achieved by our annotation support algorithm.

We measure total duration of signal segments seen by a simulated human-in-the-loop by tracking the proposal durations and potential offsets due to start and stop time corrections (we thus keep track of the *union* of proposal and ground-truth annotation). To estimate the amount of time samples seen to obtain our starting set of annotations $\mathcal{A}_{start}$, we assume that in order to obtain, for example, annotations for 10% of the total amount of events, our simulated human has looked at 10% of the total data set duration, which implicitly assumes that events are uniformly distributed across the entire data set. Our experiments work with the assumption that $\mathcal{A}_{start}$ is constructed by scrolling through recordings and "zooming in" to annotate obvious events, leading to an empty set $\mathcal{B}$, i.e., we do not require the initial annotator to explicitly label background segments as non-events.

B. Data

We use two datasets of two common biomedical signal modalities, electrocardiography (ECG) and electroencephalography (EEG), chosen for their different signal characteristics. ECG is structured and EEG is very chaotic and noisy. The ECG data set consists of lead I ECG recordings, sourced from the Computing in Cardiology 2017 Challenge [38]. We only use the (clean) sinus rhythm and atrial fibrillation recordings and superimpose artefact events on these signals, where the artefact events are taken from the Physionet MIT-BIH Noise Stress Test Database [39]. The artefacts vary in duration and are added to the background signal at different signal-to-noise ratios. The artefact events are elementwise multiplied with a Tukey window of the same size as the artefact events to ensure smooth transitions. Full details are discussed in Appendix A.

The EEG dataset consists of the Temple University Seizure Corpus [40], which contains multi-channel EEG recordings of epileptic seizures, characterized by substantial background heterogeneity and non-stationarity. The ambiguity in defining seizure start and stop times, combined with their potential confusion with signal artefacts, places this data set at the difficult end of the annotation spectrum. We use general level annotations of this data set (only annotating at which point in time a seizure occurs, not on which channel(s)), following the training and test split of [41]. We use this training set as our "to be annotated" data set. It contains 2058 seizures in 3986 recordings, spanning 713 hours of EEG. We use the test set as held-out test set for the experiment of Section III-D. It contains 673 seizures in 1013 recordings, spanning 170 hours of EEG.

Both data sets are chosen to highlight our algorithm's capabilities across varying levels of event detection complexity, from the relatively straigthforward ECG data to the highly challenging EEG data. All ground-truth annotations are available for our experiments and are used to validate our algorithm's annotation performance, ensuring a comprehensive assessment across the two extremes. Ground-truth annotations for our ECG data set indicate the actual signal events since the events are simulated. Ground-truth annotations for our EEG data are generated by the human annotators of [40] and thus might contain some amount of mistakes and inconsistencies.

C. Benchmarking

Annotation support has to save time for a human user. To evaluate the time-saving potential of our annotation support tool, we start from a randomly selected set of annotations of

about 10% of total events. We use random Bernoulli samples for every event to decide on inclusion in $\mathcal{A}_{start}$ (setting the distribution parameter to 10%). This approach mimics a scenario where an annotator randomly selects and annotates events from the dataset. We deliberately keep this initial selection mechanism simple and generic in order to focus on the algorithm's runtime aspects, effectively 'abstracting away' the initial selection process. The algorithm uses an "automated" human-in-the-loop, comparing the proposal annotations with ground-truth annotations of the data set in question. Unless stated otherwise, we produce $N = 200$ proposals per iteration for the ECG data, and $N = 500$ proposals per iteration for the EEG data (to reflect the increased difficulty of annotating epileptic seizures.)

We benchmark against the approach of [25], which despite its focus on audio processing, represents the most closely related interactive annotation system available in the literature. It identifies events in time series data and generates proposals for human verification. Unlike our method, it relies on predefined segmentation and feature extraction, limiting its adaptability to variable-duration events. Every annotated event is verified by a human, in contrast to approaches such as [26], [27]. The method of [25] segments signal recordings into short intervals, extracting hand-crafted domain-specific features from each to distinguish between event-related ("positive") and non-event ("negative") segments, similar to our sets $\mathcal{A}$ and $\mathcal{B}$. It utilizes a feature relevance scoring system to prioritize segments based on their proximity to positive features and distance from negative ones.

We run [25] for 100 iterations on ECG data and 50 on EEG data, leading to a similar computational budget as our method. For ECG artefacts, we use features sensitive to ECG artefacts: kurtosis [7], [42]–[45], skewness [7], [43], in-band to out-band power ratio [45], [46], relative power in the QRS complex [7], [44], and relative baseline power [7], together with signal power and sample entropy, extracted for 1 second windows. For EEG data, we use the state-of-the-art features of [4], covering signal statistics, spectral content, and entropy information, extracted for 2 second windows.

We count the preceding and succeeding segments when tracking signal duration seen by the human expert for the single-segment proposals of [25]. Verifying a proposal requires the signal context beyond the short single-segment boundaries. Proposed segments that match with a ground-truth event are counted as explained in Section III-A (by looking at the union of proposal and ground-truth event).

We also compare our method to exhaustive chronological search and a hypothetical ideal proposal generator to establish performance bounds.

Our annotation support algorithm finds nearly all simulated ECG events (Fig. 2a), staying close to ideal proposal generation in terms of data observed and total proposals seen by the human. Our algorithm outperfoms the method of [25], finding more events for less human effort. In terms of decisions made, we achieve around a factor four speed-up. Our algorithm and the method of [25] show a clear improvement over exhaustive chronological search.

Our annotation algorithm gives different results for epileptic seizures in EEG (Fig. 2b). The algorithm finds around 70% of the total seizure events. There seems to be a slight upwards trend left at the end. However, the algorithm generates many faulty proposals at the end, making it impractical to continue the process. We slightly outperform the method of [25] in terms of data observed and substantially in terms of total proposals. Even though both methods fail to find all events, they save human effort compared to exhaustive chronological search.

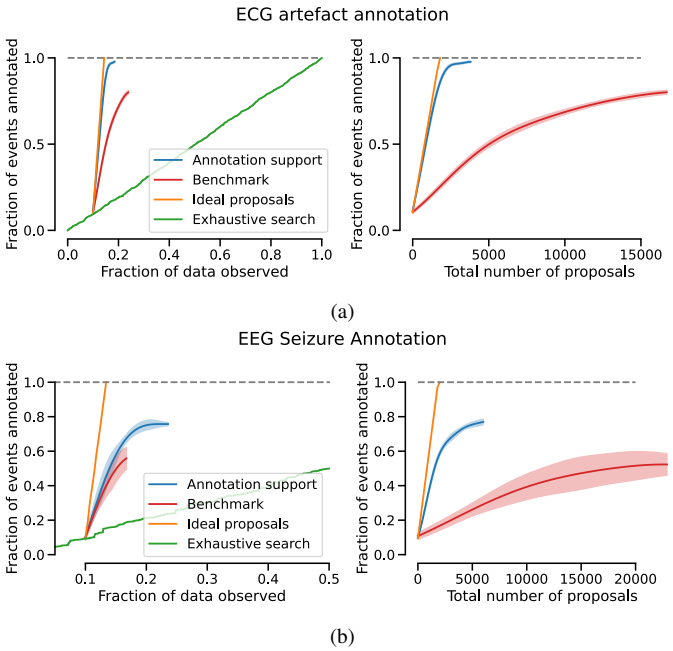


Fig. 2: Benchmark results for annotation performance. We report fraction of events found as a function of data observed by the human and the total number of proposals seen by a human. We report mean performance, with the transparent band indicating one standard deviation across 5 runs.

D. Event detection performance

We investigate how the performance of an event detection model changes with proposal-verification iterations. In practice, one might indeed be more interested in improving the performance of an event detection model by adding more labels, rather than aiming for a fully-annotated data set in itself. If an annotation support algorithm fixates on only specific subtypes of events, it can introduce bias in the set of annotations used for training. The marginal value of a single annotation can be low when a substantial portion of a data set is already annotated, or the most "informative" examples are already annotated. A human might not have to verify (or annotate) a full data set to obtain a model performing no worse than one trained on the fully annotated data set.

We evaluate generalization performance by applying the underlying event detection model (used to generate proposals) to a held-out test set. For the simulated ECG artefacts, the test set is generated from signal and noise segments not used for the training set. For EEG seizures, we use the test set of [41] as our downstream test set. At every iteration, we evaluate detection performance, reporting the area under the precision-curve with

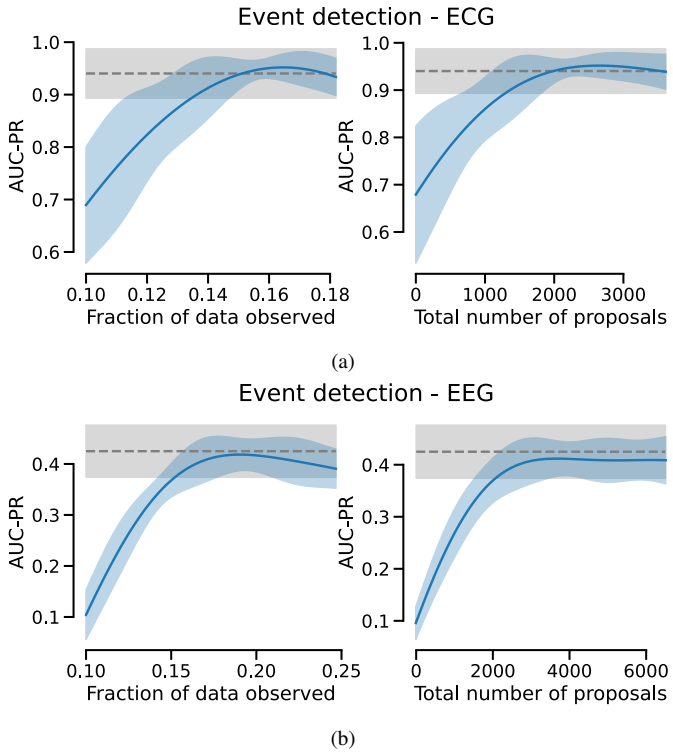


Fig. 3: Event detection performance of the ECG and EEG annotation process, showing area under the precision-recall curve (AUC-PR) for event detection on a held-out test set as a function of data observed during the annotation support process, with event detection performance evaluated at every iteration. We report mean performance, with the transparent band indicating one standard deviation across 5 runs. The grey dashed line shows mean performance of a model trained on the fully-annotated training set, with the band representing one standard deviation across 5 runs.

hits and misses of the event detection models computed as in the work of [16], using 0.5 Intersection-over-Union (IoU) as overlap threshold. Performance is compared to the detection performance of five training runs on fully-annotated data sets (using our backbone event detection model). The annotation algorithm is kept as described in the experiment above.

With 16% of the ECG data observed, a machine learning model performs on par with a model trained on the fully-annotated data set, corresponding to around 2000 proposals seen by the human-in-the-loop (Fig. 3a). Note that at 16% of data observed, the training set in question does not yet contain all training event annotations (only around 80% of events are annotated at that point). Even though the algorithm does not find all events in the EEG data (Fig. 2), test set performance is similar to using the fully-annotated data set when only observing 17% of the data set for annotation purposes (Fig. 3b). At this point, the human has been shown around 4000 proposals. Note that the base seizure detection model [16] was shown to perform similar to [13], a state-of-the-art seizure detector for the Temple University Seizure Corpus.

Crucially, for both data sets, annotating every event is not necessary to achieve event detection performance comparable to using a fully-annotated data set. Of course, this should not be viewed as a general claim, as the results depend on the specific event detection backbone used in our experiments.

E. Ablation study

We investigate three key parts of the algorithm: a) the human-in-the-loop, b) the amount of starting annotations, and c) the number of proposals per iteration.

1) Human interaction: The verification step is a substantial time investment, and should thus contribute meaningfully to the algorithm's performance. We compare the algorithm with human interaction to a version of the algorithm without human interaction. To do so, we take the algorithm setup of the previous experiments and remove the human in the loop, i.e., skip the proposal verification step. Instead, the proposal annotations $\mathcal{A}_{proposal}$ are treated as verified annotations $\mathcal{A}_{verified}$, still starting the algorithm from a small set of human-provided annotations. When this modified algorithm has met its stopping criterion, a human annotator goes through the final set of annotations $\mathcal{A}$ to discard false-positive annotations.

We measure the number of events found for similar human effort (a human either has to investigate N proposals per iteration in the human-in-the-loop setting, or investigate all proposed events when the version without human interaction finishes) and the fraction of true positives in proposals over iterations. If the algorithm properly leverages the addition of verified annotations to the set of annotations $\mathcal{A}$, we expect this true positive rate to be higher with a human in the loop, i.e., the underlying event detection model gets better at generating proposals with a human in the loop.

For ECG data, human interaction substantially improves performance, with higher true positives rates and faster event discovery. EEG data shows analogous benefits, particularly in early iterations (Fig. 4).

2) Starting number of annotations: The experiments so far all start from 10% (in expectation) of data already annotated. This entails a substantial human time cost. At the same time, if a lot of events are already annotated, finding more proposal events should become easier for the event detection model. Finding new proposal events is expected to become more difficult with fewer events annotated.

We evaluate changes to the starting annotation count (for 5 runs in each data set). We look at annotation support performance as a function of data observed, annotation performance as a function of total proposals, annotation performance as a function of iteration count, and the proposal true positive rate as a function of iterations (the latter reflects how many proposals are correct proposals). We cover a wide range of starting counts, looking at the cases where 1%, 5%, 10% (the default case), 25%, and 50 % of total events are annotated before starting our annotation support algorithm. For ECG events, the algorithm's final performance is not sensitive to the starting amount of annotations (Fig. 5a.) Even starting with 1% of events annotated, it finds nearly all events in the data set. Not surprisingly, starting with a lower amount of events increases the number of iterations until the algorithm "converges". When starting with 50% of the data annotated, the algorithm takes less than 10 iterations to find the remaining events whereas this takes 20 iterations when starting with 1% of events annotated.

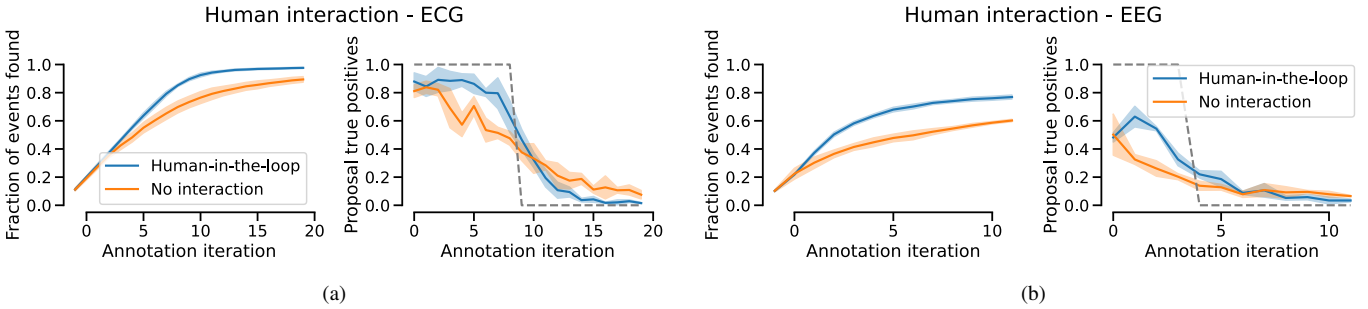


Fig. 4: Impact of human interaction when using the annotation support algorithm. Grey dashed line for the proposal true positive plots shows the behavior of the hypothetical perfect case where every proposal is correct (proposal positives fall to zero when all events are annotated). We report mean performance, with the transparent band indicating one standard deviation across 5 runs.

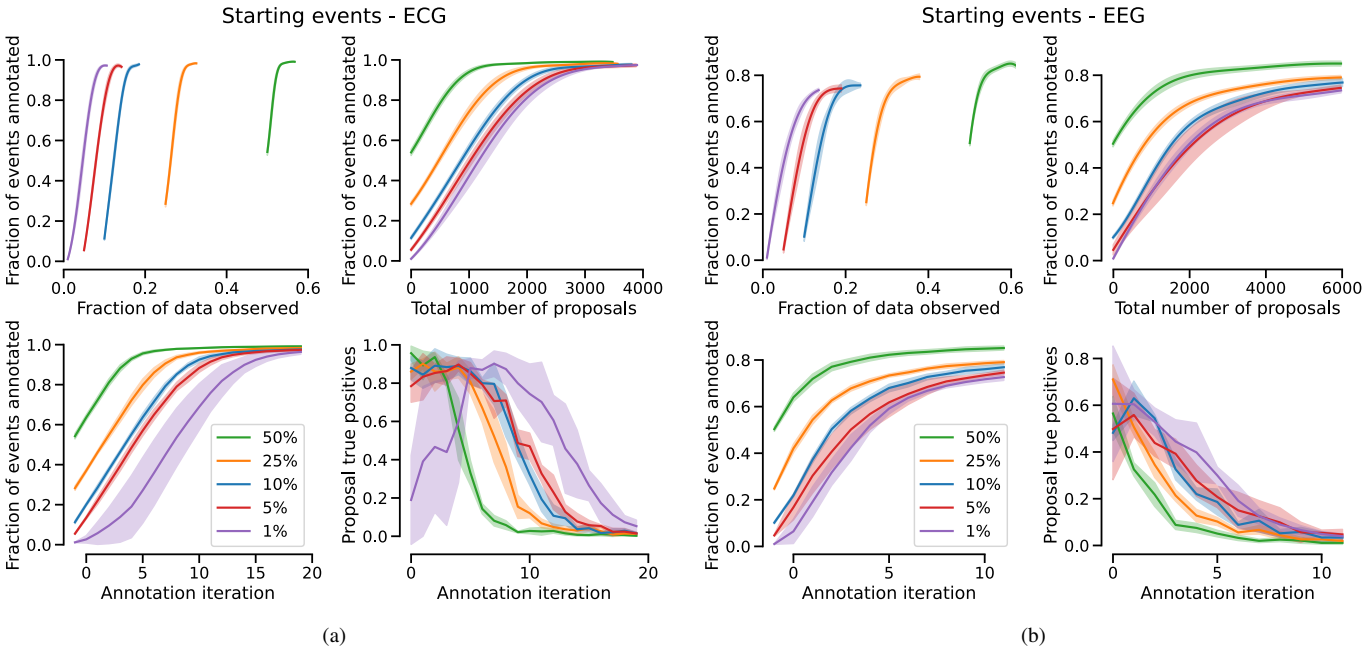


Fig. 5: Impact of varying the amount of starting annotations for the annotation support algorithm. Different colors indicate the fraction of events that are annotated at the start of the algorithm. We report mean performance, with the transparent band indicating one standard deviation across 5 runs.

While the latter takes more iterations (thereby increasing the compute time), the accumulated amount of data that has to be annotated by the human expert is much lower (<10% versus >55%). For the 1% case, however, it should be noted that for two runs out of the five, it took almost 5 iterations for the algorithm to reliably generate relevant proposals. Another interesting insight is that the slopes of these curves are similar for the different cases. The different curves are very alike, only translated depending on the number of starting events. Events as a function of data observed, number of proposals, and iterations show a similar slope for all settings, indicating that, just like the result in Fig. 2a, the algorithm is quite close to the ideal case even when starting from a low amount of events (the 1% case starts with around 20 events annotated.)

For EEG events, varying the number of starting annotations has more impact. The algorithm can get started with only 1% of events annotated, but progress is slower than when more starting annotations are available (Fig. 6b.) When less starting annotations are available, the algorithm takes longer to find more events. The different curves are not translated versions of each other anymore, as the slopes are different. Where the ECG annotation task does not benefit from more starting annotations, starting annotation count matters for the EEG annotation task.

3) Number of proposal events: Choosing the number of proposals per iteration is another hyperparameter of our algorithm. A large amount of proposals could allow a user to cover a lot of events quickly, risking a high number of "false positive" proposals and wasted effort (i.e., investigating and rejecting proposals.) A small amount, on the other hand, should result in less false positives per iteration, at the risk of a higher number of iterations to reach satisfactory performance. The total number of proposal-verification iterations has to be kept in mind when deciding on the number of proposals, as every iteration step involves training a full model on the given data set (which can be become expensive in terms of computational resources and/or compute time in between iterations).

We vary the number of proposals per iteration: 100, 200,

and 500 for ECG; 200, 500, and 1000 for EEG. Over the iteration process, we report annotation performance (expressed as annotations produced as a function of data observed and number of proposals) and how the proposal true positive rate changes with iterations.

For both data sets, the algorithm follows similar trajectories independent from the number of proposals (as seen in terms of found events as a function of expended effort) (Fig. 6.) Since the number of proposals varies, the difference lies in the number of iterations to reach the same number of annotated events, not the amount of data observed by the human. For the EEG data set the algorithm does not reach the same amount of annotations for 200 proposals as compared to the other settings. We hypothesize that this is due to the stopping criterion since all settings follow a similar trajectory. The proposal true positive rate after 12 iterations still lies around 20%, indicating that this setting can benefit from more iterations. The number of iterations has an effect on the compute time: for the same number of accumulated annotation proposals, a lower number of iterations is better as it requires less "breaks" in the human annotation process in order to recompute the model.

For ECG events, there are barely any true positive proposals past 5 iterations for 500 proposals, while around half of the proposals are still true positives for 100 proposals at 20 iterations. Note, however, that at 20 iterations when using 100 proposals, not all events are found. For EEG, a similar steep drop in true positives can be observed when using 1000 proposals, with a slower descent seen for 200 proposals. When using 200 proposals it takes many more iterations to find as many events as using 1000 proposals.

F. Multiple types of events

We investigate the impact of different types of events and their relative proportions on our algorithm's performance. So far, the experiments take an "agnostic" approach to the type of event found, in the sense that tasks only focus on distinguishing between events and background. Now, we consider the impact of different types of events present in a data set. Staying within the realm of our previous experiments, data sets can for example contain different types of artefacts and seizures with different (relative) proportions.

First, we investigate the impact of relative proportions. Our algorithm relies on patterns found in the available annotations, so we risk over-relying on a majority type. We generate ECG data sets with varying proportions of electrode motion artefacts versus muscle artefacts (both types are present in the Noise Stress Test Database[39]): 50-50, 25-75, and 10-90 splits, with the starting set of annotations reflecting these split proportions, and measure the fraction of each artefact type found by the algorithm.

Second, we investigate what happens when one is only interested in one specific type of event while other events should be ignored. We compare our base case (of Section III-C) to a setting with "distraction events", where we add an equal amount of muscle artefacts (not to be annotated) to the target electrode motion artefacts. We examine the total events found and the proposal true positive rate, with the latter expected to be impacted by the distractions (the algorithm might propose distraction events instead of the desired type).

An event type imbalance slows convergence for the minority type (in this case the electrode artefacts), but didn't prevent finding almost all events (Fig. 7.) The 90-10 split required more iterations for electrode artefacts, with a slight "ramp up" effect in earlier iterations. We observe faster convergence for the majority type (being the muscle artefacts). Distraction events had minimal impact on algorithm performance for ECG data (Fig. 8.) The number of target events found was similar to the distraction-free case, with only a sligth decrease in true positives during the first 1-2 iterations.

IV. Discussion

A. Experimental results

Our annotation support algorithm shows substantial effiency gains for ECG event detection. Starting with 1% of events annotated, we can find nearly all events by reviewing only about 10% of the data set (Fig. 6a), substantially reducing human effort compared to reviewing the entire data set (Fig. 2a). In contrast, the EEG data, characterized by its heterogeneous and non-stationary nature, presents a more challenging scenario. Here, our algorithm does not find all events (Fig. 2b), at least not without substantial effort. When we cut off the algorithm, there was still a non-zero fraction of the proposals that were actual events. Realistically speaking, however, a human annotator will likely stop annotating when 99% of proposals are wrong.

Often, the goal of annotating events is the development of event detection models. Our results indicate that, over proposal-verification iterations, we approach the performance of a model trained on a fully-annotated data set for both data sets (Fig. 3). This shows that a complete annotation is not necessary to match performance with training a model on fully-annotated data. Note, however, that in our experiments, the same model architecture was used both for annotation support and downstream event detection. Performance may vary if different models are used.

The human-in-the-loop component is crucial to our algorithm's performance, as illustrated in Fig. 4. Removing this component substantially reduces effectiveness, even in a simpler task like ECG event detection. While fully automating our annotation support algorithm (by removing the human-in-the-loop) is not impossible, it does present considerable challenges. Without human verification, the algorithm tends to generate numerous events, including many false positives, across both simple (ECG) and complex (EEG) data sets. Therefore, it is advised to involve a human expert for verifying intermediate rounds of proposals, as intended in our algorithm design.

To use our annotation support algorithm, initial starting annotations are required. The algorithm performs well with only a few starting events for a more simple task, but more complex events benefit from more initial annotations. Nonetheless, even for challenging events the algorithm can also start from few annotations (Fig. 5), but achieving statisfactory performance may take more iterations in this case.

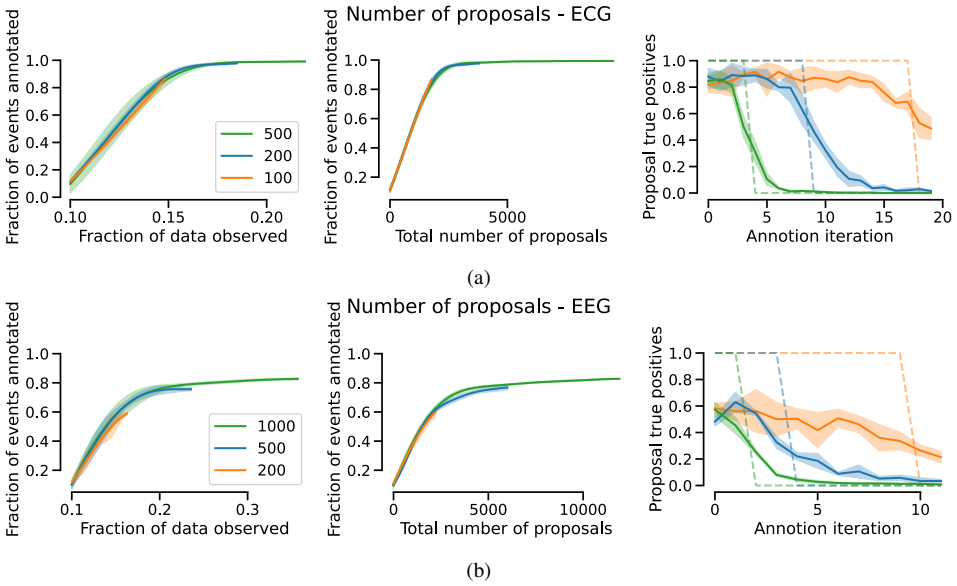


Fig. 6: Impact of the number of proposals when using the annotation support algorithm. Dashed lines for the proposal true positives shows the behavior of the hypothetical perfect case where every proposal for the corresponding number of proposals is correct. We report mean performance, with the transparent band indicating one standard deviation across 5 runs.

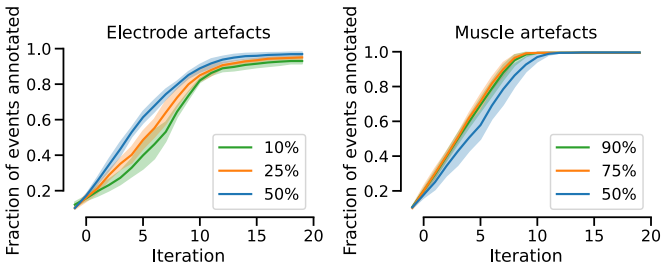


Fig. 7: Impact of event type proportions on algorithm performance. Data sets consist of both types of events with the shown proportion splits. We report mean performance, with the transparent band indicating one standard deviation across 5 runs.

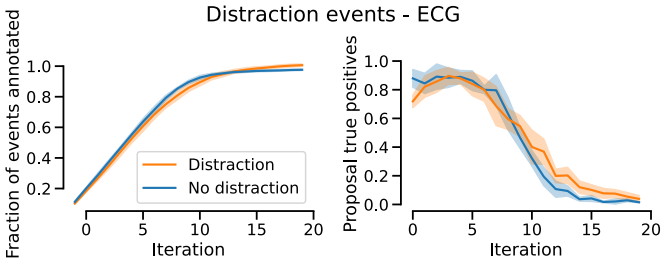


Fig. 8: Impact of "distraction" events on algorithm performance. We report mean performance, with the transparent band indicating one standard deviation across 5 runs.

The number of proposals per iteration mainly impacts the required number of iterations to achieve desired performance levels, but it does not impact the total human effort nor the performance of the algorithm (Fig. 6). However, the perceived effort can differ substantially between verifying 100 proposals versus 1000 proposals per iteration (even if the total number of proposals remains the same across all iterations). We also made abstraction of the fact that every iteration involves a pause for training a new model, which can affect the total annotation time. The numbers we tested are substantial fractions of the total number of events. Both data sets contain around 2000 events, and we use 200 proposals (10% of total events) for the simpler ECG events and 500 proposals (25% of total events) for the more difficult EEG events. For new tasks, expert knowledge can help decide on a good number of proposals to control the number of iterations.

Our algorithm is robust to handling different types of events, performing well both in mixed settings (Fig. 7) and when focusing on one event type while another is also present (Fig. 8). However, extreme imbalances between event types require caution. In our ECG data set, a 90-10 split does not lead to breakdown of the algorithm, but more substantial imbalances or tougher conditions could lead to different results. Indeed, detecting the minority event type in our 90-10 split starts out more slowly. Greater imbalances might lead to worse results.

B. Using and extending the annotation support algorithm

For our experiments, we use of a generic underlying event detection algorithm [16], albeit with a task-specific backbone model. Using our full annotation support algorithm would involve deciding on an underlying event detection model. It is possible to use the model of [16] for new tasks with minimal effort, as it is fairly generic and removes the need for post-processing-based event detection algorithms. Nonetheless, it is reasonable to expect that the performance of our annotation support algorithm will depend on the underlying event detector. If good models exist for the task at hand, it can be worthwhile to incorporate them *if they are robust against "false negative" event annotations* (events that are not yet annotated.) It is key that the event detection model can still train in the presence of false negatives, otherwise the proposal generation process will not work. The model of [16] exhibits this property.

For other event detection models in the literature, it is unknown whether they can train in this false-negative setting.

Our annotation support algorithm is optimized for datasets where the number of events is manageable by human experts, which is not necessarily related to the raw size of the dataset. It performs well even with low event prevalence, as demonstrated in our experiments on EEG seizure data. However, it is less suitable for extremely large datasets where the volume of events surpasses what human verifiers can realistically handle. In such cases, the human verification process could become impractical, limiting the applicability of our method.

Novel applications of our annotation support algorithm will likely fall somewhere between the two settings investigated in our experiments. The ECG and EEG data sets are representative for the two extreme cases. Finding new annotations in the basic ECG data set is a relatively easy task, as evidenced by the close adherence to the "ideal case" during our experiments (Fig. 2a). On the other hand, learning to detect seizure events is by itself a very difficult task, as evidenced by the relatively poor performance of state-of-the-art seizure detection algorithms, even if they are tailored to the specific data set used in our study [13], [47]–[49]. Depending on the complexity of a novel task, the algorithm's behavior is expected to be more like the ECG case or the EEG case.

Despite its simple design, our proposed algorithm shows good results. A priori, it might seem unlikely that our proposal generation process inspired by pseudo-labeling can achieve such strong results. And yet, for simple events we show that our proposal generation process is close to optimal (Fig. 2a.) Nonetheless, there is room for improvement when annotating more difficult events. The proposal generation process can be changed, which is something we do not investigate in this paper but can be interesting as future work. Proposal generation will, however, be limited by state-of-the-art performance for the underlying event detection task. For example, seizure detection is a task where state-of-the-art algorithms are not perfect [41]. Improving the proposal generation process is mainly relevant for known difficult events. Our simple proposal generation process performs well for simpler events.

One limitation of the presented study is our proxy for human effort. As explained before, there will be a strong relation between the amount of data seen or number of decision made by the annotator, and actual time spent on the annotation, but this relation is not perfect. Nonetheless, we stand by the conclusion that the proposed annotation support algorithm can be a serious time saver for annotating data sets. Even if our proxy for human effort is too optimistic, there is substantial margin compared to exhaustively going through our experiment data sets (as seen in Fig. 2).

Our algorithm lacks mechanisms to handle false positive annotations by the human annotator, assuming all verified proposals are true events. This limitation can impact proposal generation effectiveness. While multiple expert annotators could mitigate risks, our implementation doesn't detect or correct false positives during annotation. The impact of annotation mistakes is difficult to determine, as mistakes can be understandable (annotating spurious patterns that are similar to an event) or severe (confusing a clear background segment for an event), which will differently impact the result. As this paper represents an initial step towards flexible annotation support in biomedical signals, addressing false positives is considered to be future work to enhance the algorithm's reliability and practical applicability.

Future work in annotation support for biomedical signals should focus on these two critical enhancements: evaluating the algorithm with actual human annotators to gain real-world insights into effectiveness and usability, and developing robust methods to handle false-positive annotations. These advancements will address current limitations, improving reliability and practical applicability across various biomedical signal processing tasks.

Our experiments as presented assume an unambiguous ground truth. Unfortunately, expert disagreement is common in biomedical signal annotation [50], [51]. Care should be taken when multiple experts are employed for the verification step. Modifications will depend on how the human experts will cooperate. Our algorithm requires a singular decision for every proposal, whether to accept or reject it. Experts can split proposals among themselves, all review the full set of proposals, or pick a setup in between. Disagreement can be settled by majority voting, or by consensus. Each of these decisions will, most likely, impact the patterns and biases our algorithm picks up over the annotation process.

We evaluate and discuss our algorithm mainly in terms of human effort, but the computational aspect is important to mention. The time spent for our experiments is mainly driven by the proposal generation process, since we simulate the human-in-the-loop. For every iteration, the proposal generation process involves training a deep learning model on the entire data set, which is expensive in terms of computing resources. For most biomedical signal processing tasks, the human experts will be the key limiting factor. At the same time, compute cost should not be ignored. The total investment will be the combination of human time spent and computational time. A single training run for proposal generation in the case of our EEG data takes four hours on an NVIDIA RTX 2080 GPU. If proposals can be verified in comparable or less time, the compute time does become an important factor.

V. Conclusion

In this paper, we propose an annotation support algorithm. The algorithm keeps a human in the loop by iterating between proposal generation and verification steps. We demonstrate on two data sets that the algorithm allows for substantial savings in terms of human effort when annotating events in biomedical signals. Task-specific design aspects are limited to an underlying event detection model. Changes to this underlying model allow for straightforward extensions for annotating other data set with other modalities and/or event types than those used in this study.

References

[1] 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.

[2] A. Mincholé *et al.*, "Machine learning in the electrocardiogram," *Journal of electrocardiology*, vol. 57, S61–S64, 2019.

[3] K. Rasheed *et al.*, "Machine learning for predicting epileptic seizures using EEG signals: A review," *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 139–155, 2020.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] B. Hunyadi *et al.*, "Incorporating structural information from the multichannel EEG improves patient-specific seizure detection," *Clinical Neurophysiology*, vol. 123, no. 12, pp. 2352–2361, 2012.

[9] S. J. Redmond and C. Heneghan, "Cardiorespiratory-based sleep staging in subjects with obstructive sleep apnea," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 3, pp. 485–496, 2006.

[10] E. Alickovic and A. Subasi, "Ensemble SVM method for automatic sleep stage classification," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 6, pp. 1258–1265, 2018.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] N. Seeuws *et al.*, *Avoiding post-processing with event-based detection in biomedical signals*, 2023. arXiv: 2209.11007 [eess.SP].

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

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

[19] L. Zhao *et al.*, "Quantitative signal quality assessment for large-scale continuous scalp EEG from a big data perspective," *Physiological Measurement*, 2022.

[20] A. Malafeev *et al.*, "Automatic artefact detection in single-channel sleep EEG recordings," *Journal of sleep research*, vol. 28, no. 2, e12679, 2019.

[21] 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.

[22] N. Seeuws *et al.*, "Electrocardiogram quality assessment using unsupervised deep learning," *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 2, pp. 882–893, 2021.

[23] J. Macea *et al.*, "In-hospital and home-based long-term monitoring of focal epilepsy with a wearable electroencephalographic device: Diagnostic yield and user experience," *Epilepsia*, vol. 64, no. 4, pp. 937–950, 2023.

[24] D. van der Wal *et al.*, "Biological data annotation via a human-augmenting AI-based labeling system," *NPJ digital medicine*, vol. 4, no. 1, p. 145, 2021.

[25] B. Kim and B. Pardo, "A human-in-the-loop system for sound event detection and annotation," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 8, no. 2, pp. 1–23, 2018.

[26] L. Cao *et al.*, "Smile: A system to support machine learning on EEG data at scale," *Proceedings of the VLDB Endowment*, vol. 12, no. 12, pp. 2230–2241, 2019.

[27] M. Nashaat *et al.*, "Asterisk: Generating large training datasets with automatic active supervision," *ACM Transactions on Data Science*, vol. 1, no. 2, pp. 1–25, 2020.

[28] Y. Zhao *et al.*, "An EEG annotation system facilitating brain disease research," in *2023 16th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, IEEE, 2023, pp. 1–6.

[29] V. Gerla *et al.*, "Expert-in-the-loop learning for sleep EEG data," in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, IEEE, 2018, pp. 2590–2596.

[30] J. Freitas *et al.*, "Confidence in the qualified crowd: A platform for sourcing EEG annotations," in *2020 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, IEEE, 2020, pp. 1–6.

[31] A. Bhardwaj *et al.*, "Human-in-the-loop rule discovery for micropost event detection," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[32] M. Nalisnik *et al.*, "An interactive learning framework for scalable classification of pathology images," in *2015 IEEE International Conference on Big Data (Big Data)*, IEEE, 2015, pp. 928–935.

[33] B. Settles, "Active learning literature survey," 2009.

[34] A. P. Dempster *et al.*, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society: series B (methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[35] V. Losing *et al.*, "Incremental on-line learning: A review and comparison of state of the art algorithms," *Neurocomputing*, vol. 275, pp. 1261–1274, 2018.

[36] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine learning*, vol. 109, no. 2, pp. 373–440, 2020.

[37] N. Boukhelifa *et al.*, "Evaluation of interactive machine learning systems," *Human and machine learning: visible, explainable, trustworthy and transparent*, pp. 341–360, 2018.

[38] 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.

[39] G. B. Moody *et al.*, "A noise stress test for arrhythmia detectors," *Computers in cardiology*, vol. 11, no. 3, pp. 381–384, 1984.

[40] V. Shah *et al.*, "The Temple University hospital seizure detection corpus," *Frontiers in neuroinformatics*, vol. 12, p. 83, 2018.

[41] N. Neurotech and NeuroTechX. "Neureka™ 2020 Epilepsy Challenge." (), [Online]. Available: https://neureka-challenge.com/ (visited on 06/12/2020).

[42] T. He *et al.*, "Application of independent component analysis in removing artefacts from the electrocardiogram," *Neural Computing & Applications*, vol. 15, pp. 105–116, 2006.

[43] G. Clifford *et al.*, "Signal quality indices and data fusion for determining clinical acceptability of electrocardiograms," *Physiological measurement*, vol. 33, no. 9, p. 1419, 2012.

[44] Q. Li *et al.*, "Robust heart rate estimation from multiple asynchronous noisy sources using signal quality indices and a Kalman filter," *Physiological measurement*, vol. 29, no. 1, p. 15, 2007.

[45] T. H. Falk, M. Maier, *et al.*, "MS-QI: A modulation spectrum-based ecg quality index for telehealth applications," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 8, pp. 1613–1622, 2014.

[46] G. D. Clifford *et al.*, "ECG statistics, noise, artifacts, and missing data," *Advanced methods and tools for ECG data analysis*, vol. 6, p. 18, 2006.

[47] A. Thyagachandran *et al.*, "Seizure detection using time delay neural networks and LSTMs," in *2020 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, 2020, pp. 1–5. DOI: 10.1109/SPMB50085.2020.9353636.

[48] J. Pedoeem *et al.*, "TABS: Transformer based seizure detection," in *2020 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, 2020, pp. 1–6. DOI: 10.1109/SPMB50085.2020.9353612.

[49] L. Wei and C. Mooney, "Epileptic seizure detection in clinical EEGs using an XGboost-based method," in *2020 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, 2020, pp. 1–6. DOI: 10.1109/SPMB50085.2020.9353625.

[50] S. R. Benbadis *et al.*, "Interrater reliability of EEG-video monitoring," *Neurology*, vol. 73, no. 11, pp. 843–846, 2009.

[51] A. C. Grant *et al.*, "EEG interpretation reliability and interpreter confidence: A large single-center study," *Epilepsy & Behavior*, vol. 32, pp. 102–107, 2014.

[52] Z. Liu *et al.*, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 976–11 986.

[53] J. Schlemper *et al.*, "Attention gated networks: Learning to leverage salient regions in medical images," *Medical image analysis*, vol. 53, pp. 197–207, 2019.

TABLE I: ECG event detection architecture. Decoder stages are indicated as ()'. All stages consist of ConvNeXt [52] convolution blocks, with the exception of stage 0 which is a single convolution operation. Upsampling is performed with a normal upsampling operation. Downsampling is performed using a strided convolution with kernel size and stride equal to the downsampling factor. Encoder and decoder stages are merged in a U-Net-like fashion.

Stage	# Filters	Kernel size	# Blocks	Downsampling factor
0	32	15	N/A	1
1	32	15	3	4
2	64	15	3	16
3	128	7	3	64
4	128	5	5	256
5	128	3	3	1024
4'	128	5	5	256
3'	128	7	3	64
2'	64	15	3	16
Output	2	7	N/A	16

APPENDIX

A. Simulated data generation

ECG events are generated using the approach of [16]. Signal segments of 20 s duration are drawn from the Computing in Cardiology 2017 Challenge data set [38] with strides of 5 s. We only use the recordings from the normal rhythm or atrial fibrillation classes to avoid artefact events that are not introduced by our data generation process.

1) Base ECG data set: For our base ECG data set, artefact events from the Noise Stress Test data set [39] are added in randomly into the ECG segments with 20% chance. The event generation process happens once for every ECG segment, and the resulting data (signals with events mixed in, combined with annotation targets) are then used as finite data sets for our experiments. When artefact events are mixed in, one or two artefact segments of random duration are taken from the database. Durations of the artefact segments are randomly distributed between 1 s and 6.7 s. These artefact events are added to the ECG signal with a signal-to-noise ratio (where artefact events are considered as noise) chosen uniformly random between -6 and 6.

2) Mix of event types: The ECG data set consisting of mixes of event types follows the same process. Every artefact event is randomly drawn from the electrode motion artefacts or muscle artefacts. This decision is a Bernoulli process, with "success probability" set to the desired mix proportion.

B. Model architectures

1) ECG Events: The ECG detection architecture is a U-Net-like architecture built for event-based modeling [16] (producing a center and duration signal.) Table I shows the backbone architecture. For training, the model uses the Adam optimizer with 0.0001 as learning rate for 50 epochs. During training and testing, the model uses 20 s segments as input.

2) EEG Events: The EEG detection architecture follows the EEG model of [16], with an additional downsampling step to account for more signal context. Table II shows the backbone architecture. For training, the model uses the Adam optimizer with 0.0001 as learning rate for 50 epochs. During training, the model uses input segments of 200 s. During testing, the model uses full recordings (by convolving over the full recording.)

TABLE II: Seizure detection architecture. Stage 4 and 4' are connected in a U-Net-like fashion, concatenating features of stage 4 and upsampled features of stage 5'. Stages 5 and 5' are connected using upsampled features of stage 6. Stages use convolution layers with the given hyper-parameters. The ()' stages have two convolution-normalization-nonlinearity blocks. Decoder stages merge channel-level information using Attention Gating [53]. Encoder stages are channel-independent (using the same convolution filters on every channel). Stage 6 consists of one channel-independent convolution block, max-pooling over the EEG channels, and two convolution blocks with dropout.

Stage	# Filters	Kernel size	Downsampling factor
0	16	15	1
1	32	15	4
2	64	15	16
3	64	7	64
4	128	5	256
5	128	5	1024
6	128	5	4096
5'	64	5	1024
4'	64	5	256
Output	2	1	256

C. Implementation details

The annotation support algorithm was implemented using PyTorch. All experiments were run on an NVIDIA RTX 2080 GPU. For the ECG task, each iteration of proposal generation took approximately half an hour, while for the EEG task, it took approximately four hours.