Efficient and precise annotation of local structures in data

Efficient and precise annotation of local structures in data

by John Martinsson

LUND
UNIVERSITY

Licentiate Thesis
Thesis advisors: Maria Sandsten, Olof Mogren
Faculty opponent: Annamaria Mesaros

To be presented, with the permission of the LTH of Lund University, for public criticism in the room MH:309A at the Centre for Mathematical Sciences, Mathematical Statistics on Thursday, the 3rd of October 2024 at 10:15.

Organization
LUND UNIVERSITY

Centre for Mathematical Sciences, Mathematical Statistics
Box 118
SE–221 00 LUND, Sweden

Author(s)
John Martinsson

Document name
Licentiate thesis

Date of presentation
2024-10-03

Sponsoring organization The Swedish Foundation for Strategic Research (SSF; FID20-0028), and Sweden's Innovation Agency (2023-01486)

Title and subtitle
Efficient and precise annotation of local structures in data:

Abstract
Machine learning models are used to help scientists analyze large amounts of data across all fields of science. These models become better with more data and larger models mainly through supervised learning. Both supervised learning and model validation benefit from annotated datasets where the annotations are of high quality. A key challenge is to annotate the amount of data that is needed to train large machine learning models. This is because annotation is a costly process and the collected labels can vary in quality. Methods that enable cheap annotation of high quality are therefore needed.

In this thesis we consider ways to reduce the annotation cost and improve the label quality when annotating local structures in data. An example of a local structure is a sound event in an audio recording, or a visual object in an image. By automatically detecting the boundaries of these structures we allow the annotator to focus on the task of assigning a textual description to the local structure within those boundaries. In this setting we analyze the limits of a commonly used annotation method and compare that to an oracle method, which acts as an upper bound on what can be achieved. Further, we propose new ways to perform this kind of annotation that results in higher label quality for the studied datasets at a reduced cost. Finally, we study ways to reduce annotation cost by making the most use of each annotation that is given through better modelling approaches in general.

Key words
Annotation efficiency, Machine learning, Sound event detection

Classification system and/or index terms (if any)

Supplementary bibliographical information

Language
English

ISSN and key title
1404-028X

ISBN
978-91-8104-199-6 (print)
978-91-8104-200-9 (pdf)

Recipient's notes

Number of pages
120

Price

Security classification

I, the undersigned, being the copyright owner of the abstract of the above-mentioned dissertation, hereby grant to all reference sources the permission to publish and disseminate the abstract of the above-mentioned dissertation.

Signature _____ Date 2024-09-12

DOKUMENTDATABLAD enl SIS 61 41 21

Efficient and precise annotation of local structures in data

by John Martinsson

LUND
UNIVERSITY

Funding information: The Swedish Foundation for Strategic Research (SSF; FID20-0028), and Sweden's Innovation Agency (2023-01486)

Mathematical Statistics
Centre for Mathematical Sciences
Lund University
Box 118
SE-221 00 Lund
Sweden

www.maths.lth.se

Licentiate Thesis in Mathematical Sciences 2024:3
ISSN: 1404-028X

ISBN: 978-91-8104-199-6 (print)
ISBN: 978-91-8104-200-9 (pdf)
LUTFMS-2020-2024

© John Martinsson, 2024

Printed in Sweden by Media-Tryck, Lund University, Lund 2024

Till Lara

Abstract

Machine learning models are used to help scientists analyze large amounts of data across all fields of science. These models become better with more data and larger models mainly through supervised learning. Both supervised learning and model validation benefit from annotated datasets where the annotations are of high quality. A key challenge is to annotate the amount of data that is needed to train large machine learning models. This is because annotation is a costly process and the collected labels can vary in quality. Methods that enable cheap annotation of high quality are therefore needed.

In this thesis we consider ways to reduce the annotation cost and improve the label quality when annotating local structures in data. An example of a local structure is a sound event in an audio recording, or a visual object in an image. By automatically detecting the boundaries of these structures we allow the annotator to focus on the task of assigning a textual description to the local structure within those boundaries. In this setting we analyze the limits of a commonly used annotation method and compare that to an oracle method, which acts as an upper bound on what can be achieved. Further, we propose new ways to perform this kind of annotation that results in higher label quality for the studied datasets at a reduced cost. Finally, we study ways to reduce annotation cost by making the most use of each annotation that is given through better modelling approaches in general.

Publications

Publications concerning the work of this thesis have been made as follows:

A **Modelling the annotation quality and cost of weak labeling of fixed length segments in audio data**
John Martinsson, Olof Mogren, Tuomas Virtanen, Maria Sandsten
Unpublished manuscript

B **From weak to strong sound event labels using adaptive change-point detection and active learning**
John Martinsson, Olof Mogren, Maria Sandsten, Tuomas Virtanen
32nd European Signal Processing Conference (EUSIPCO), Lyon, France, 2024. (Nominated for best student paper.)

C **DMEL: the differentiable log-Mel spectrogram as a trainable layer in neural networks**
John Martinsson, Maria Sandsten
ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, South Korea, 2024.

D **Few-shot bioacoustic event detection using an event-length adapted ensemble of prototypical networks**
John Martinsson, Martin Willbo, Aleksis Pirinen, Olof Mogren, Maria Sandsten
Proceedings of the 7th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2022), Nancy, France, 2022, pp. 121-125.

.

Acknowledgements

Firstly, I would like to thank both of my supervisors Olof Mogren and Maria Sandsten. Thank you for creating an academic environment where I feel safe to explore research questions that interest me without the fear of failure, and for your encouragement to do so. Secondly, I would like to thank Tuomas Virtanen for allowing me to visit his research group, and for a very nice ongoing collaboration around topics that have ended up shaping this thesis.

I would also like to thank my colleagues Edvin Listo Zec and Martin Willbo for making our work place a fun and engaging place to be, and for always engaging in scientific discussions.

Finally, I would like to thank Lara. Thank you for always being there for me, and showing your support even in times when you should be focusing on yourself and our future family. I am not very good at celebrating the small achievements in life, but you always show me how. I love you.

Funding

Funding for this research comes from the Swedish Foundation for Strategic Research (SSF; FID20-0028), Sweden's Innovation Agency (2023-01486), and the AI Center at RISE Research Institutes of Sweden.

Popular summary

Machine learning is at the core of the recent success of artificial intelligence. A machine learning model is a form of computer algorithm that learns from data. The most common and reliable way to get the model to learn is by providing supervision. This is done by feeding the model with input data, say an audio recording, and then telling the model how to describe what is in that audio recording. A description of an audio recording given by a model is called a prediction, and a description given by a human that tells the model what to predict is called a label.

As an example, we can feed an audio recording of a bird singing into such a model and then tell it to predict that there is bird singing in the recording by providing an audio recording with an appropriate label. We can do this many times with audio recordings of different animal vocalizations such as dogs barking, whales calling, or birds singing. Eventually the model will learn to predict what is in a given audio recording.

These predictions can then be used to automatically analyze large amounts of recorded audio data to gain new scientific insights and establish policies. As an example, we could detect the vocalizations of different animal species in an acoustically monitored habitat to better understand the biodiversity in that habitat, and to establish policies towards maintaining or improving the biodiversity.

At the core of supervised learning is the human description of the data, the label. If the label is of low quality, for example by indicating that a bird is singing in a part of the audio recording where it is not, then the predictions from the trained model will be of low quality.

What we explore in this research are ways to improve supervised training of machine learning models by helping the human annotator to provide labels of higher quality. The goal is higher quality predictions at a reduced annotation cost. Resulting in higher quality scientific insights and policies, at a reduced cost.

Contents

Introduction

My interest in using machine learning to detect animal vocalizations in audio recordings goes back to a hiking trip in the Himalayas in 2016. During the month-long hike in the mountains my uncle would tell me the names of some different species of birds that were singing along the trail. I had recently come into contact with neural networks, and started to develop the idea that this should be possible to do with a computer. This idea never left me, and when I came home I discovered that there is a whole field with researchers doing just this called *bioacoustics*. This became the topic of my master's thesis, where I classified bird song in audio recordings using convolutional neural networks. And five years later I started my doctoral studies on the topic of machine learning for audio analysis, often called *machine listening*.

At the core of machine listening is the detection and classification of sound events. Sound events are distinct sounds that we can identify and recall based on their descriptions. These events, like "bird singing" or "dogs barking", form the core elements of a sound scene, helping us to interpret the surrounding environment. First we need to notice the onset and offset of the sound event (detection), and then we need to describe the type of event that has occurred (classification). Textual descriptions of these sound events are typically short, capturing the essence of what we hear. When a human is performing the detection and classification we call it annotation, and in this thesis we will call the textual description a class label, and the onset and offset a segment label. Annotated sound events are necessary to train and evaluate machine listening models that perform sound event detection. The speed of annotation can vary depending on the annotation task, and the annotations are inherently subjective, influenced by the annotator's personal experiences and perceptions [1].

What I have learned while developing methods and by discussing and collaborating with ecologists from around the world, is that large scale audio datasets are scarce. Datasets with annotations of thousands of animal sound events are rarely available when these projects start. They often have very few, if any, labeled sound events, and a huge need to annotate months or years of unlabeled audio recordings. What I want to explore in this thesis is therefore ways to make annotation easier, and how to improve the quality of the labels

collected during annotation. Further, I explore ways to make the best use the few initial labels that we may have.

In Paper A, we develop a theory for the label quality and annotation cost of a commonly used labeling method, where the annotator is limited to assigning class labels to fixed length audio data segments, called FIX weak labeling.

In Paper B, we propose a weak labeling method where the annotator assigns class labels to data segments that are adapted to cover the local structures (sound events) of interest. This can save annotation cost by requiring the annotator to give class labels to fewer data segments, and can also make the labels more precise by adapting the data segments to the structures of interest.

In Paper C, we propose a method to learn the time-frequency resolution in the commonly used log-Mel spectrogram as a part of the neural network training process.

In Paper D, we propose a robust method for bioacoustic sound event detection which can learn from only five annotated sound events.

While the motivation for this research mainly comes out of the need for cost efficient and high quality annotations for audio data, a lot of the research may be applicable to other types of data as well. The theory developed, and the methods proposed can in principle be applied to any time series data, and could possibly be extended to annotation of data in 2 or 3 dimensions as well (e.g., images or point clouds). I will therefore talk more broadly about annotation of data with local structures, such as sound events, in this thesis (see chapter 1).

The thesis is divided into three main chapters. First, chapter 1 gives an introduction to annotation of local structures in data and explains in more detail what we mean by a local structure. This chapter puts Paper A and Paper B in perspective. Second, chapter 2 gives an introduction to machine guided annotation of local structures in data, and puts Paper B, Paper C and Paper D in perspective. Finally, chapter 3 conclude the thesis, and discuss interesting future research directions.

Chapter 1

Annotating local structures in data

In this chapter, I will introduce the concept of a local structure in data, why we want to annotate local structures, and what it means to do so. I will then describe a common method for annotating local structures without explicitly asking the annotator to describe the boundaries of the structures. We will compare this method to an oracle method that uses the knowledge of the true boundaries and quantify the gap between them.

The oracle method can be seen as an upper bound on what may be achievable if we were to use the properties of the local structure during the annotation process. Finally, I will discuss some of these properties, and how we may be able to use them to design more precise annotation methods, which will lead us into the next chapter.

1 What is a local structure?

A local structure is a local part of the data that a group of people have given a textual description to. In figure 1.1 a local structure (green) is illustrated for audio (left) and image data (right). In the audio example, the local structure is the sound event (green) associated with the textual description ”bird song”. In the image example, the local structure is the set of pixels that make up the visual object (green) associated with the textual description ”bird”. The other parts of the data (gray) illustrate the other things happening in the background.

An important property of a local structure is that it occurs locally. In the audio example the sound event occurs locally along the time dimension, and in the image example the visual object occurs locally along the spatial dimensions. For audio, the local occurrence is typically associated with some form of temporal coherence where dependencies between previous and future sound samples are strong.

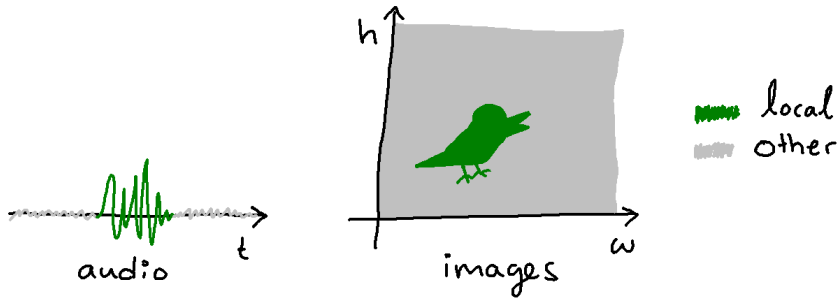


Figure 1.1: An example of a local structure in audio and image data. The audio data is a recording of a bird singing, and the image data is a picture of a bird. The local structures are shown in green. In the audio recording it is the sound event of the bird singing, and in the image data it is the visual object of the bird.

2 Why and how do we annotate local structures?

In supervised learning we train the model to predict the labels given by the annotator, and during evaluation we choose the model that best predict the labels. The prediction quality of the model will therefore necessarily depend on the quality of the annotations. We have seen this in sound event detection, where precise labels for the sound events lead to better performing models [2].

In extent, the (scientific) insights that can be drawn from the model predictions depend on the quality of the model. As an example, in ecology a researcher may be interested in counting the number of animal vocalizations from a certain species in an audio recording. The number of vocalizations, used together with a model of vocalization frequency, can then be used to estimate the number of individuals in a recording [3]. The accuracy of this count will depend on the quality of the annotations.

For evaluation data, a higher label quality means a better specification of what the best model should do, which is always desirable. E.g., if the goal is a model that produces well detected onsets and offsets of sound events, then the labels of the evaluation data need to reflect this. However, when training machine learning models, label noise can act as a form of regularization during training, meaning that sometimes noisy training labels can actually result in a model that generalize better to the evaluation data. Despite this, I will argue that anything that can be achieved with noisy training labels can also be achieved with noise free training labels by simply adding the noise afterwards. The opposite is not true. From this perspective, less noisy labels are strictly better also for the training data.

This is why we want to annotate the local structure; a precise local structure annotation gives a more accurate description of the data that help us develop better machine learning models.

Annotation of a local structure require us to describe the boundaries of the local structure, and to give a textual description of the structure within those boundaries. We therefore consider two label categories: the *segment label* (boundaries) and the *class label* (textual description). The segment label describes the boundaries of a data segment, and the class label is the textual description that we attribute to that data segment.

Labeling data with local structures therefore consists of constructing a set of segment labels and their corresponding class labels. The set of segment labels should partition the data into disjoint segments that cover all of the data, meaning that every part of the data is associated with exactly one class label. In figure 1.2 we show two different sets of segment labels leading to a correctly (top) and incorrectly (bottom) labeled local structure in an audio recording.

In the top image of figure 1.2 we can see that assigning the correct class label to each of the three segments will result in a correctly labeled local structure in the audio recording. In the bottom image, however, we can see that it is impossible to assign a correct class label to the second segment since it covers two different classes of the data, the local structure (green) and background sounds (gray). Further, we can see that three is the minimum number of segments needed to correctly label the audio data in this case, since if we have any fewer we will necessarily have to cover both the green and gray part with one of them. However, there are many ways, using more segments, that would also result in correct labels.

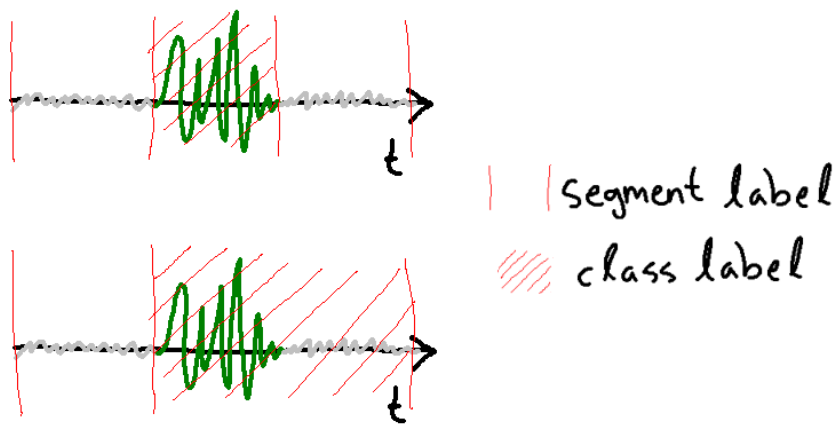


Figure 1.2: An example of segment labels and class labels of a local structure in audio data. The local structure is shown in green. The top image shows a way to partition the audio data into correct segment labels. With correct segment labels we can attribute a class label to each segment without introducing any label errors. In this example the class label indicates the presence of the local structure. The bottom image shows a way to create incorrect segment labels, in this case we will necessarily introduce noise into the labels by assigning a class label to each segment.

We will refer to the data point that we want to annotate, e.g., an audio recording in this example, as x and the label of x as y. The label $y = (s, c)$ consists of a set of segment labels s and a set of corresponding class labels c. We are interested in understanding and reducing

the label noise introduced by incorrect segment labels in this thesis, called *segment label noise*. A segment label is incorrect if it covers data from multiple classes, because then there is no correct class label for that segment. Segment label noise has been shown to lead to decreased performance of sound event tagging models [4, 5], where the goal is to detect if a sound event occurs in a given audio recording.

Another type of label noise occurs when an annotator assigns the incorrect class label for a given data segment, we call this *class label noise*. We do not model class label noise in Paper A, but we do study the effect of it in Paper B. A common way to reduce class label noise is to form a consensus on the class label by asking multiple annotators to label the same data segment [6, 7, 8].

3 Weak labeling of local structures

Annotating local structures is a demanding task that requires the annotator to detail the boundaries for the segment labels and assign the correct class label to each of these segments. The segment labels given by annotators are often inconsistent [2], partly because the interpretation of what constitutes the boundary of a local structure is subjective [9], leading to segment label noise. In addition, annotation of segment labels is demanding and takes more time, increasing the cost of annotation, and if the annotator is not an expert they may misunderstand the annotation task if it is too complex [8]. All these challenges associated with getting segment labels of high quality from annotators has created a need for methods that do not explicitly ask the annotator for the segment labels.

We therefore consider the setting where we only ask the annotator for a class labels of a given data segment, called weak labeling. This means that the segment labels need to be automatically constructed. The automatic construction of labels is often called pseudo-labeling [10]. In this thesis, we are interested in understanding the segment label noise resulting from the weak labeling process, which is a form of pseudo-labeling, and we propose ways to reduce this noise to get more precise annotation of the data.

3.1 FIX and ORC weak labeling

A commonly used weak labeling method is to partition the data into fixed and equal length segments, we will call this FIX weak labeling. The annotator is then asked to provide class labels for the data within each segment, and the corresponding segment label is inferred from the boundaries of the segment. The FIX weak labeling method is illustrated in figure 1.3. For audio data, this means that the annotator assigns class labels to equal sized audio segments. For images, the class labels would be assigned to rectangle segments, and

so on.

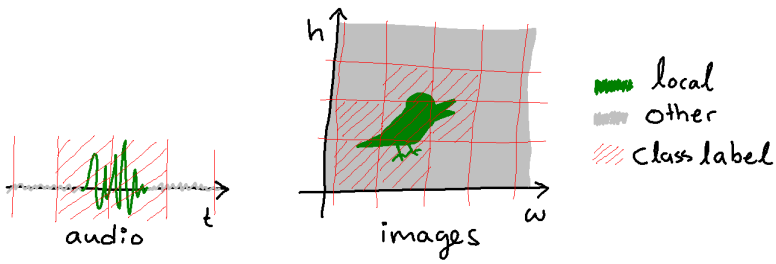


Figure 1.3: The FIX weak labeling method for audio (left) and images (right). The data is partitioned into equally fixed sized segments, and the annotator is asked to assign a class label to these segments. In these examples we can see how FIX introduces false positives, the parts of the class label that do not overlap with the local structure.

Variations of FIX weak labeling have been used to collect many of the large scale audio datasets that exist today. Annotation of AudioSet [11], which is still one of the largest and most used audio datasets today, was done by selecting a subset of 10 second segments to be weakly annotated. A more recently collected dataset called MAESTRO Real [8] was annotated by asking for class labels of 10 second segments with 9 second overlap to reduce segment label noise, and by asking five annotators to annotate each segment to reduce class label noise. They end up with multiple class labels for each part of the audio data and perform a weighed majority vote, based on an estimate of annotator competence, to get a single label for each part of the data.

The segment size as well as the overlap has an effect on the segment and class label noise. Too small segments may result in the annotator missing the presence of a local structure, which can introduce class label noise, but too large segments will introduce segment label noise. In general, smaller segments and larger overlap also mean that the annotator has to assign more class labels which increase the annotation cost. Which segment size and overlap to choose therefore depends on assumptions of the annotators' ability to detect the local structures. For some local structures the annotator may have to observe the whole structure to give a correct class label.

In Paper A we develop a theory for the limits of FIX weak labeling in 1 dimension (e.g., audio). We restrict ourselves to the setting where there is no overlap between the segments, and study the effect that the annotator model has on the resulting segment label noise for varying segment sizes. We introduce a metric called query intersection over union (QIoU), and an intuitive way to think of this metric is that $1 - \text{QIoU}$ roughly correspond to the segment label noise for a given segment. Using this we develop an expression for the segment size that will minimize the segment label noise in expectation for a given annotator model and data distribution.

We compare to an oracle (ORC) weak labeling method, which should be considered as

an upper bound that we can not know in practice. The ORC weak labeling method is illustrated in figure 1.4. It asks the annotator to assign a class label to the ground truth local structures. By construction this method will never introduce segment label noise, it will also always ask the annotator for the fewest possible number of class labels (three in the audio example, and two in the image example).

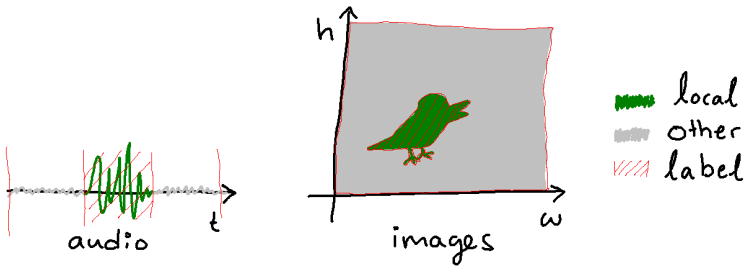


Figure 1.4: The ORC weak labeling method for audio (left) and images (right). There are no false positives, and we ask for the fewest number of class labels, 3 in the audio case, and 2 in the image case.

It may seem counterintuitive to call this ORC *weak* labeling, since this method gets labels that are of equal quality as the best strong labeling method where the class labels and segment labels are given by an oracle annotator. However, it is a weak labeling method in the sense that it only asks the annotator for class labels and never for segment labels. Thus acting as an upper bound on weak labeling, which we can not know, but that we can try to estimate in practice.

This highlights the potential of modelling the ORC weak labeling method by exploiting properties of the local structures, such as local similarities, dependencies and coherence, to automatically construct the segment labels, which is what we do in Paper B. In Paper B we use these properties to create segments that are adapted to the sound events of interest, and show that this can lead to higher quality labels at a reduced annotation cost. This is called machine guided annotation, which is the topic of the next chapter.

Chapter 2

Machine guided annotation of local structures in data

"The real problem is what can man and machine do together and not in competition."— Richard W. Hamming

In this chapter we will look at machine guided ways to reduce the annotation cost of local structures in data. We will introduce the annotation loop and the different ways to reduce annotation cost. Then we will connect Paper B-D to these, and finally discuss some differences of our proposed method and other methods.

1 The data annotation loop

We consider the setting where the annotator can only provide class labels for given data segments. The boundaries of the local structures therefore need to be automatically estimated, and then the annotator is asked to attribute a class label to the data segment contained within the boundaries.

Let us start with the generic annotation loop given in Algorithm 1. The algorithm takes as input a set of unlabeled data $\mathcal{D}_U = \{x_i\}_{i=1}^n$, a model M, and a performance criterion τ. The algorithm iteratively samples the next data point x to be annotated from the set of unlabeled data points, asks an annotator to label that data point, and then updates the model using the new annotation. The new model performance τ_M is then evaluated, and we continue this annotation loop until a satisfactory model performance τ has been reached. The output of the algorithm is the set of m labeled data points $\mathcal{D}_L = \{(x_i, y_i)\}_{i=1}^m$, that

results in a model M of satisfactory performance $\tau_M \geq \tau$. The set of unlabeled data points $\mathcal{D}_U$ that we want to annotate can for example be a set of audio recordings, or a set of images such as those illustrated in figure 1.1 in the previous chapter, and the goal is a label y for each data point x that describes the local structures in it.

Algorithm 1 Annotation of data

1: **Input:** Unlabeled data $\mathcal{D}_U = \{x_i\}_{i=1}^n$, model M, performance criterion τ
2: **Output:** Labeled data $\mathcal{D}_L$
3: $\mathcal{D}_L \leftarrow \emptyset$
4: $\tau_M \leftarrow$ evaluate model M
5: **while** $\tau_M < \tau$ **do**
6: sample and remove x from unlabeled data $\mathcal{D}_U$
7: annotator gives label y to x and adds (x, y) to labeled data $\mathcal{D}_L$
8: update model M using labeled data $\mathcal{D}_L$
9: evaluate the model to get performance τ_M
10: **end while**
11: **return** $\mathcal{D}_L$

We are interested in ways to reduce the total annotation cost to get a model M with performance $\tau_M \geq \tau$. There are many ways to achieve this goal, each focusing on a separate line in Algorithm 1.

Firstly, looking at line 6, we can sample the next data point x such that the gain in model performance is maximized when x is annotated (using knowledge of M). This is the goal of works in active learning [12, 13].

Secondly, looking at line 7, we can either reduce the annotation cost c associated with the annotator giving the label y to x, or we can improve the quality of the annotation y. A higher quality annotation should lead to a higher gain in model performance. In Paper B we propose a method to increase the label quality of y for a given x at a reduced annotation cost c. We do this by using the model M to guide the annotator towards a higher quality y.

Lastly, looking at line 8, we can design models M that gain more in performance from each update. In Paper D we propose a few-shot learning method which is designed to learn a lot from only a few annotated examples, and in Paper C we propose a differentiable log-Mel spectrogram (DMEL) that can be optimized jointly with the model M.

2 Increasing the label quality at a reduced annotation cost

Increasing the quality of the label y given by the annotator for data point x can be done by increasing the labeling capability of the annotator. This can be done, for example, by choosing an expert annotator, or increasing the annotators' ability to perform the task [14]. Both these are ways of changing the properties of the human annotator, which we will not consider here.

We can also guide the annotator during the annotation task in ways that facilitate higher quality. This can be done, for example, by providing better annotation interfaces [15], or by doing parts of the annotation work automatically [16, 17, 18]. Automating parts of the annotation work has the benefit that label quality can potentially be increased at the same time as the annotation cost is reduced. In Paper B we propose at a weak labeling strategy towards this end.

Let us consider the annotation cost associated with assigning a label y to a data point x (line 7 in Algorithm 1). As described in section 2, the label $y = (s, c)$ consists of a set of segment labels $s = \{s_1, \ldots, s_k\}$ that partition the data point x (e.g., an audio recording) into k disjoint data segments that completely cover x, and a set $c = \{c_1, \ldots, c_k\}$ of the k corresponding class labels given by the annotator. The partitioning of x into k disjoint segments need to be done automatically since we are restricted to only ask the annotator for class labels. The annotation cost can therefore be written as ck where c is the cost of assigning a class label to a data segment. If we need to annotate m data points to achieve model performance τ the total cost therefore becomes mkc. We can reduce this cost by reducing any of the three factors in the product. We will consider m and k in this thesis, as c is a property of the human annotator.

The number of needed annotations can be reduced if the quality of the labels is increased. The quality of the label y can be affected by class label noise and segment label noise (see section 2). Let $Q(x, y)$ be a measure of the quality of the label y given to x with respect to the true class labels and local structures. Let $\bar{Q} = \frac{1}{n} \sum_{(x,y)\in\mathcal{D}_L} Q(x, y)$ denote the average label quality of the annotated dataset $\mathcal{D}_L$ resulting from Algorithm 1.

In Paper B we propose a method that makes use of the model M to partition x into segments that are better adapted to the local structures of interest. We call this machine guided annotation, because the model M is used to guide the annotation towards higher quality segment labels. Further, the model is updated with each new annotation. Initially, this may lead to noisy segment labels, but as the model is updated this noise is reduced. We empirically show that this happens, and that it leads to a higher label quality on average for the same annotation budget k compared to other commonly used methods. An improved average label quality $\bar{Q}$ means that the number of annotation rounds m needed to reach model performance τ is effectively reduced.

Automating parts of the annotation work does come with certain risks. If the automatically constructed segment labels contain a lot of segment label noise, then we may end up with lower quality labels instead. We have not observed this to be a problem in Paper B, but it is important to be aware of this risk. However, a nice property of first constructing the segments and then asking the annotator to give them class labels is that if the segments are noisy, then the annotator can notice this and take appropriate actions.

There is a subtle difference in this setup to other recently proposed pseudo-labeling methods for time series data, where the weak labels are given *before* the pseudo-labeling. In [16, 18] the weak label is first collected for a given point in time and then propagated to cover the local structure according to a temporal coherence criterion, and in [17] the weak labels are used to train a machine learning model which then predicts the pseudo-labels for the local structure, and then another model is trained on the pseudo-labels.

In our setup, by performing the weak-labeling *after* the pseudo-labeling we make sure that the annotator looks at the pseudo-labels, giving a natural quality assurance to the labeling process.

3 Learning more from the annotations

Broadly speaking, this is the goal of most work in supervised machine learning. We want to develop models that learn well from annotated training data, meaning that they generalize to some annotated evaluation data. However, there are specialized research directions such as few-shot learning, where the goal is to learn well from very few training annotations [19, 20, 21, 22]. Few-shot learning methods are, however, typically not designed to scale with more annotations. So, there is a trade-off here, and which way to model the data depends on the budget you have for annotation. If the budget is very low you may consider few-shot learning methods such as the one explored in Paper D, and if the budget is reasonably large then you may consider more complex ways of modelling such as that explored in Paper C.

To realize the ideas in Paper B we need good ways of modelling audio data in general. In Paper D we look at ways to make the most use of a few annotations. We do this by using an event length adapted ensemble of prototypical neural networks [19]. The key idea in the paper is to choose embedding functions for the ensemble that have been trained for certain event lengths based on the event lengths of the few examples that we already have. In Paper C we propose a version of the log-Mel spectrogram where the window length of the underlying short-time Fourier transform can be optimized jointly with the neural network model. The window length defines the resolution in time and frequency of the log-Mel spectrogram, and optimizing this for the classification task at hand can lead to stronger models. The log-Mel spectrogram is a very commonly used input representation

for convolutional neural networks (CNNs) in audio.

4 The difference between active learning and active annotation

In principle, the active updating of the model with each new annotation proposed in Paper B fits into the general framework of *active learning*, which is the reason active learning is in the title of the paper. However, I now believe that it may be reasonable to distinguish between them, and would like to propose the term *active annotation* for machine guided annotation where the model is iteratively updated during the annotation loop.

In active learning, we typically consider the setting where the next data point x is sampled to maximize some uncertainty criteria of the model (line 6 in Algorithm 1 depends on M). The idea is that the data point x that the model is most uncertain about should be most informative to annotate next, and that by biasing the data sampling process in this way we can reduce the number of annotations needed to reach a satisfactory model performance. The label noise is assumed independent of the data point to annotate.

In active annotation, the annotator is guided by the model M during the annotation of a *given data point* x. This means that the label noise will depend on the data point to annotate.

That is, active learning is about changing the sampling of x using knowledge of M, active annotation is about changing the sampling of y given x using knowledge of M.

Chapter 3

Conclusions and future work

1 Conclusions

We have developed a theory for weak labeling of local structures in data measured along 1 dimension, such as time series or audio data, and studied the limits of an approach commonly used in practice. We have compared this to an oracle method that solves the weak labeling task optimally. Knowing the consequences of different choices when performing weak labeling is crucial to make sure that the resulting annotations are of sufficient quality. (Paper A)

The limits frame the weak labeling problem, and can be used to put current methods in context. Further, the developed theory may give insights into ways to develop improved weak labeling methods. Towards this end we have also developed a weak labeling method that aims to model the oracle method by using each new annotation to further improve the annotation quality through machine guidance. We have showed that this method of annotation results in a higher label quality on average on all the studied datasets and for all assumed annotator models. (Paper B)

We have also proposed a method to learn the resolution in time and frequency of the typically used log-Mel spectrogram in audio modelling. We have showed that learning the appropriate resolution for the task at hand as a part of model training can speed up the training process, and lead to better performing models. (Paper C)

Finally, we have explored a modelling method that only requires as few as five annotated local structures to perform well, and have proposed two ways of improving the robustness of that method towards problems where the local structures vary a lot in size. (Paper D)

2 Future work

The papers that have shaped this thesis the most are Paper A and Paper B, which are about annotation of data with local structures, and in particular sound data. There are many future research direction that could be explored on this topic.

2.1 FIX weak labeling in more than 1 dimension

In Paper A we derive a theory for FIX weak labeling of local structures that appear along 1 dimension of the data, such as in time series data. It would be interesting to extend this theory into D dimensions, or at least into 2 and 3 dimensions such as images or point clouds. We rarely annotate in more than 3 dimensions anyway.

The number of class label assignments needed should grow exponentially with D for the studied FIX weak labeling method, but linearly with the number of local structures in the data for ORC weak labeling, making the potential cost gain of adaptive methods higher in more dimensions. Of course, modelling the ORC weak labeling method will also become a harder task with more dimensions. Exploring what happens when more dimensions are introduced is a very interesting research direction.

2.2 Active learning and active annotation in combination

While active learning is about changing the sampling of the data point x using the model, active annotation is about changing the sampling of the label y *given* a data point x using the model.

Clearly there will be a tension between these two processes if they are used jointly.

The best model we can hope to learn is a perfect model of the annotator, meaning that a sample that is hard for the model should also be hard for the annotator to annotate, leading to more label noise. By studying active annotation and active learning jointly, we may gain new insights into this trade-off between label noise and hardness of sample and find that the best way to learn may not be to always be exposed to the hardest sample of the problem, but rather a reasonably hard sample. The question is: what is reasonably hard?

2.3 Model selection in the active annotation loop

The underlying model in Paper B is a prototypical neural network [19], which was developed to learn quickly from only a few annotations. Unfortunately, the way this is done also means

that the learning saturates rather quickly. We have seen this in experiments where the label quality on a held out test set (not part of the paper) saturated after annotation of around 20 to 50 audio recordings.

Note that the label quality after saturation is better than that of the methods we compare with, so it is still beneficial to use this approach. But, it would be even better if the label quality just kept increasing until we eventually learn to model the ORC weak labeling process. We will probably not reach this upper bound, but we should aim to.

A very interesting research direction would be to make the complexity of the model depend on the number of annotations available through some model selection criteria. For example, we have seen in other works on active few-shot learning [21] that a prototypical neural network can have better performance than a linear classifier applied on the same embeddings when only a few annotations are available, but that the linear classifier becomes better after a certain number of annotations. The question is when to make the switch from the simple model (prototypical neural network) to the more complex model (a linear model applied on the embeddings), and going further when to choose models with even higher capacity as we accumulate more annotations.

2.4 Other annotator models

In Paper A we derive the theory for an annotator model that can detect presence of local structures if a fraction $\gamma \in (0, 1]$ of the local structure is contained within a given segment. While this makes sense for some types of sound events, other assumptions may be more applicable for other types of data.

In general, a better understanding of these properties of human annotators in practice would be very beneficial, and empirical studies towards this end are encouraged.

2.5 Adaptive weak labeling of multiple classes

The adaptive weak labeling method proposed in Paper B is developed for presence or absence annotation of a certain sound event class of interest. That is, to annotate multiple classes we need to perform multiple binary annotation tasks. However, the underlying prototypical neural network should be fairly easy to extend to multiple sound event classes of interest to facilitate annotation of multiple classes in a single annotation pass. There are trade-offs between multi-pass binary annotation and single-pass multi-label annotation [23], and being able to choose between these would be beneficial.

References

[1] Annamaria Mesaros, Toni Heittola, and Dan Ellis. Datasets and evaluation. In Tuomas Virtanen, Mark Plumbley, and Dan Ellis, editors, *Computational Analysis of Sound Scenes and Events*, chapter 6, pages 147–179. Springer Cham, 2017.

[2] Shawn Hershey, Daniel P.W. Ellis, Eduardo Fonseca, Aren Jansen, Caroline Liu, R. Channing Moore, and Manoj Plakal. The benefit of temporally-strong labels in audio event classification. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 366–370, 2021. ISSN 15206149. doi: 10.1109/ICASSP39728.2021.9414579.

[3] Tiago A. Marques, Len Thomas, Stephen W. Martin, David K. Mellinger, Jessica A. Ward, David J. Moretti, Danielle Harris, and Peter L. Tyack. Estimating animal population density using passive acoustics. *Biological Reviews*, 88(2):287–309, 2013. doi: https://doi.org/10.1111/brv.12001. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/brv.12001`.

[4] Ankit Shah, Anurag Kumar, Alexander G. Hauptmann, and Bhiksha Raj. A closer look at weak label learning for audio events, 2018. URL `https://arxiv.org/abs/1804.09288`.

[5] Nicolas Turpault, Romain Serizel, Emmanuel Vincent, Nicolas Turpault, Romain Serizel, and Emmanuel Vincent. Analysis of weak labels for sound event tagging. 2021. URL `https://hal.inria.fr/hal-03203692`.

[6] Irene Martin-Morato, Manu Harju, and Annamaria Mesaros. Crowdsourcing Strong Labels for Sound Event Detection. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 246–250, 2021. ISSN 19471629. doi: 10.1109/WASPAA52581.2021.9632761.

[7] Irene Martín-Morató, Manu Harju, Paul Ahokas, and Annamaria Mesaros. Training Sound Event Detection with Soft Labels from Crowdsourced Annotations. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2023. ISSN 15206149. doi: 10.1109/ICASSP49357.2023.10095504.

[8] Irene Martin-Morato and Annamaria Mesaros. Strong Labeling of Sound Events Using Crowdsourced Weak Labels and Annotator Competence Estimation. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 31:902–914, 2023. ISSN 23299304. doi: 10.1109/TASLP.2022.3233468.

[9] Anurag Kumar and Bhiksha Raj. Deep cnn framework for audio event recognition using weakly labeled web data, 2022. URL `https://arxiv.org/abs/1707.02530`.

[10] Eric Arazo, DIego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Pseudo-Labeling and Confirmation Bias in Deep Semi-Supervised Learning. *Proceedings of the International Joint Conference on Neural Networks*, 2020. doi: 10.1109/IJCNN48605.2020.9207304.

[11] Jort F. Gemmeke, Daniel P.W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio Set: An ontology and human-labeled dataset for audio events. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 776–780, 2017. ISSN 15206149. doi: 10.1109/ICASSP.2017.7952261.

[12] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[13] Steve Hanneke. Theory of Active Learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309, 2014.

[14] Meng Fang, Xingquan Zhu, Bin Li, Wei Ding, and Xindong Wu. Self-Taught Active Learning from crowds. *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 858–863, 2012. ISSN 15504786. doi: 10.1109/ICDM.2012.64.

[15] Mark Cartwright, Ayanna Seals, Justin Salamon, Alex Williams, Stefanie Mikloska, Duncan MacConnell, Edith Law, Juan P. Bello, and Oded Nov. Seeing sound: Investigating the effects of visualizations and complexity on crowdsourced audio annotations. *Proceedings of the ACM on Human-Computer Interaction*, 1(CSCW):–246, 2017. ISSN 25730142. doi: 10.1145/3134664.

[16] Yooju Shin, Susik Yoon, Sundong Kim, Hwanjun Song, Jae Gil Lee, and Byung Suk Lee. Coherence-Based Label Propagation Over Time Series for Accelerated Active Learning. *ICLR 2022 - 10th International Conference on Learning Representations*, 2022.

[17] Heinrich Dinkel, Zhiyong Yan, Yongqing Wang, Junbo Zhang, and Yujun Wang. Pseudo Strong Labels for Large Scale Weakly Supervised Audio Tagging. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2022-May:336–340, 2022. ISSN 15206149. doi: 10.1109/ICASSP43922.2022.9746431.

[18] Di Chen, Xin-Yi Li, Ang Li, and Yu-Bin Yang. Representation-Based Time Series Label Propagation for Active Learning. *2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 1154–1159, 2023. doi: 10.1109/cscwd57460.2023.10152835.

[19] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, pages 4078–4088, 2017. ISSN 10495258.

[20] Yu Wang, Justin Salamon, Nicholas J. Bryan, and Juan Pablo Bello. Few-shot sound event detection. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2020-May:81–85, 2020. ISSN 15206149. doi: 10.1109/ICASSP40776.2020.9054708.

[21] Yu Wang, Mark Cartwright, and Juan Pablo Bello. Active Few-Shot Learning for Sound Event Detection. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 1551–1555, 2022. ISSN 19909772. doi: 10.21437/Interspeech.2022-10907.

[22] Ines Nolasco, Shubhr Singh, Veronica Morfi, Vincent Lostanlen, Ariana Strandburg-Peshkin, Ester Vidaña-Vila, Lisa Gill, Hanna Pamuła, Helen Whitehead, Ivan Kiskin, Frants H. Jensen, Joe Morford, Michael G. Emmerson, Elisabetta Versace, Emily Grout, Haohe Liu, Burooj Ghani, and Dan Stowell. Learning to detect an animal sound from five examples. *Ecological Informatics*, 77(May), 2023. ISSN 15749541. doi: 10.1016/j.ecoinf.2023.102258.

[23] Mark Cartwright, Graham Dove, Ana Elisa Méndez Méndez, Juan P. Bello, and Oded Nov. Crowdsourcing Multi-label Audio Annotation Tasks with Citizen Scientists. *Conference on Human Factors in Computing Systems - Proceedings*, pages 1–11, 2019. doi: 10.1145/3290605.3300522.

Scientific publications

Author contributions

Co-authors are abbreviated as follows: Maria Sandsten (MS), Olof Mogren (OM), Tuomas Virtanen (TV), Martin Willbo (MW), and Aleksis Pirinen (AP).

Note that ideas for papers almost never come from one person alone, they are mostly a collaborative effort even when we do not realize. Therefore, when I write that I came up with the idea for a paper, I mean that I rather independently thought about the problem to be solved and came up with the main parts of the idea for the methods to try and the experiments and simulations to be done. If a person is a co-author, they probably contributed to the idea of the paper in one way or another.

Paper A: Modelling the annotation quality and cost of weak labeling of fixed length segments in audio data

I came up with the idea for the paper, constructed the proofs to derive the theoretical results, implemented the simulation experiments, and wrote the manuscript (including figures and tables). All co-authors (MS, OM and TV) have provided valuable feedback during the development of the manuscript, including feedback on presentation of proofs and the design of the simulations.

Paper B: From weak to strong sound event labels using adaptive change-point detection and active learning

I came up with the idea for the paper, and developed the idea together with TV during a research visit at his group. Further, I have conducted all the simulations and experiments for the paper under weekly supervision of TV, and I have written the paper (including

figures and tables). All co-authors (MS, OM and TV) have provided valuable feedback during the development of the method and the paper.

Paper C: DMEL: the differentiable log-Mel spectrogram as a trainable layer in neural networks

MS and I came up with the idea for the paper. I have conducted all the simulations and experiments, and I have written the majority of the paper including tables and figures. MS have developed the theory section in the paper, and provided very valuable guidance and insights during the development of the method.

Paper D: Few-shot bioacoustic event detection using an event-length adapted ensemble of prototypical networks

The ideas in the paper were a collaborative effort between all co-authors. I implemented the ideas, performed the training and evaluation of the model, derived all results, and wrote the most of the paper. All co-authors (OM, MS, MW, and AP) have provided feedback during the development of the method and paper.

Paper A

Modelling the annotation quality and cost of weak labeling of fixed length segments in audio data

John Martinsson[1,2], Olof Mogren[1], Tuomas Virtanen[3], Maria Sandsten[2]

[1] *Computer Science, RISE Research Institutes of Sweden*

[2] *Centre for Mathematical Sciences, Lund University*

[3] *Signal Processing Research Centre, Tampere University*

Abstract

Training and evaluating machine learning models for sound event detection (SED) require data with sound event annotations. A common way to annotate sound events in audio recordings is to present fixed length segments (FIX) to an annotator who gives each segment a class label. In order to better understand the limits of FIX labeling we model the label quality and derive an expression for the associated cost. We consider an annotator model that assigns a class label to a given segment if the segment covers a large enough fraction of the event. We compare FIX labeling with oracle (ORC) labeling that uses the ground truth event timings to derive the segments for annotation, and we provide a thorough theoretical analysis which quantifies the gap between the two methods. We show that the label quality of the FIX method decreases with stricter annotator criteria while ORC always give perfect labels. We further show that the cost of optimal FIX labeling grows linearly with the length of the audio recording to annotate, while the cost of ORC grows linearly with the number of events occurring in the recording. Our work provides theoretical grounds for making informed decisions about the annotation process in sequence labelling tasks, and a motivation for some of the recently proposed works that improves over FIX in both cost and performance.

Proceedings: Unpublished manuscript

Keywords: Active learning, annotation, sound event detection, deep learning

1 Introduction

Training and evaluating sound event detection (SED) models require audio data with event labels, a description of which type of sounds that are occurring at different times in an audio recording [1].

Sound events are distinct sounds that we can identify and recall based on their descriptions. These events, like people talking or dogs barking, form the core elements of a sound scene, helping us to interpret the environment around us. Class labels for these sound events are typically short and descriptive, capturing the essence of what we hear, while the onset and offset labels describe where the sound event starts and ends in time. When it comes to annotating these sounds, manual annotation involves human annotators mapping audio content to textual labels. The speed of annotation can vary depending on the annotation task, and the annotations are inherently subjective, influenced by the annotator's personal experiences and perceptions [2].

Crowdsourcing is a promising method for obtaining sound event annotations for large datasets. Using platforms like Zooniverse, crowdsourcing can quickly gather numerous judgments, producing high-quality annotations through inter-annotator agreement measures [3]. Crowdsourcing is particularly suitable when the annotation task is broken down into multiple easy to perform unit tasks [4]. The reason is that when crowdsourcing annotations we do not always know the expertise of the annotators, and the annotation task therefore needs to be easy to understand and preferably easy to perform. This has been shown to lead to more consistent and higher quality labels [3]. Easier annotation tasks can also lead to happier and more motivated annotators due to the decreased level of effort and skill needed to complete the task [5].

A way to break multi-class sound event annotation into many smaller unit tasks is to perform a binary annotation round for each sound event class. The trade-off between one round of multi-label annotation and multiple rounds of binary annotation has been studied by Cartwright et al. [6]. To further simplify the annotation task, we can ask the annotator only to indicate the presence of a certain sound event in a given audio segment, called weak labeling. The onset and offset of the event is then instead inferred from the timings of the audio segment.

Binary weak labeling is attractive because the task is easy to understand and perform by annotators, and thus suited for crowdsourcing annotations. This is the annotation setting we consider in this work. The annotator is asked to recognize the presence of a given sound event class in a given audio segment, and when no presence is detected the audio segments is given the absence label.

An attractive way to collect binary weak labels in practice is to segment the audio recording

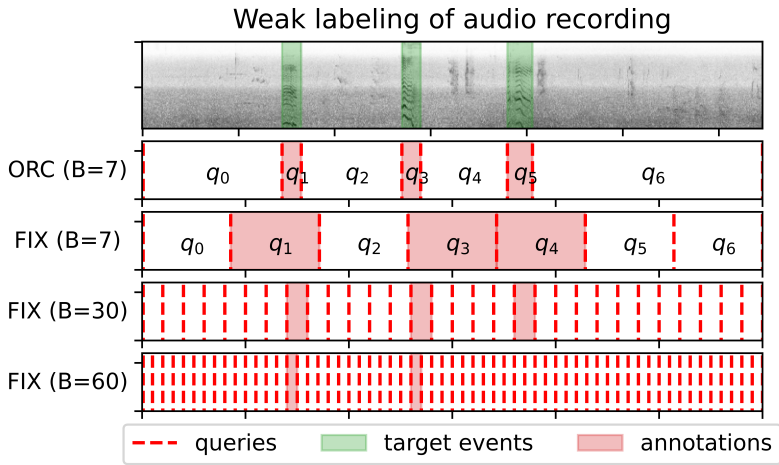


Figure 1: Annotating an audio recording with three target events (green) using the ORC and FIX method with different budgets B. The annotator is asked to annotate each query segment q_i (omitted from figure for $B = 30$ and $B = 60$) with either the presence (red) or absence (white) of the target event. The event labels from ORC are by definition optimal, while the event labels from FIX vary in quality with the number of query segments B chosen.

into fixed length audio segments (FIX) [4, 3]. We will refer to this labeling process as FIX labeling.

In general, two types of sound event label errors can occur: (i) incorrect class labels, and (ii) incorrect onsets and offsets for the temporal activity of the sound events. The latter type of label error is often a problem when performing FIX labeling.

Most audio datasets today consists of data collected using weak labeling [7]. The performance of SED models, however, is known to improve with event labels that contain precise onsets and offsets [8]. This becomes especially important when we want to count the number of occurrences of an event class. For example in bioacoustics, where counting the number of vocalizations of an animal species can be used to estimate population density and draw ecological insights [9].

FIX labeling is also commonly used in active learning for sound event detection. The audio data is divided into equal length query segments which are then chosen according to some criterion. The chosen segment is then given a class label by an annotator [10, 11, 12].

We need to better understand the limits on label quality for FIX labeling, and the consequences of choosing different segment lengths. This will allow us to make informed choices on the segment length, and understand how and why to develop alternative annotation method [13] which try to model ORC labeling.

Figure 1 shows the presence annotations (red) for an audio recording using either ORC or FIX labeling with different number of query segments B. We query the annotator for a class

label for each segment, and will often refer to these segments as simply a query. The top panel shows the spectrogram for the audio recording with the target events shown in green. The minimum number of queries needed to get event labels that are perfectly localized in time using ORC is $B = 7$ in this case.

The annotations from FIX vary in quality depending on the number of queries used. The presence labels for FIX can contain a lot of absence sounds ($B = 7$), and we may need many query segments to see that there are at least three events ($B = 30$) occurring. Further, if the annotator need to hear a fraction of the total event to detect its presence, then FIX will miss events when we have too many query segments ($B = 60$). Here we have assumed that that the annotator need to hear at least 50% of the entire event to detect presence. The annotator therefore misses the third event when $B = 60$ since none of the query segments now overlap with 50% or more of the event.

Correct onsets and offsets are not only important for training SED models, but also for evaluating them. If our goal is a SED model that detect the temporal activations of events well then the labels for the evaluation data must reflect this goal. Otherwise, we may end up rejecting models that has learned to do what we want (e.g, models that can learn well from incorrect onsets and offsets) in favor of models that do not (e.g., models that exactly learns to predict the incorrect onsets and offsets).

Consider the event labels in figure 1, and assume that we only have access to labels from FIX ($B = 7$) labeling. Further, assume that we have one model that can perfectly detect the temporal activations of the events, and one that produce predictions exactly as the FIX ($B = 7$) labels. Since our evaluation labels are incorrect we will have to reject the correct model in favor of the incorrect model.

The goal of this paper is to better understand the gap in annotation quality and annotation cost between FIX and ORC labeling under different annotator models and audio recording distributions. To this end we introduce a metric for measuring the label quality of a query segment which we call query intersection over union (QIoU). We derive an expression for the expected QIoU for FIX labeling under certain assumptions and for different annotator models. We also provide an expression for the number of query segments, B, needed to achieve the highest QIoU in expectation using FIX.

We quantify the trade-off between the number of segments (B) and the quality of the event labels (QIoU) for FIX. For example, in figure 1 the cost of having an annotator answer $B = 7$ queries is lower than the same annotator answering $B = 30$ queries, but the resulting labels for $B = 7$ are also worse.

The theory can be used to understand which segment length to use for FIX to get the best label quality and the cost of this choice. And the insights may facilitate further development of annotation methods [13] that try to model the ORC labeling process to get stronger event

labels.

2 Related work

In this section we will describe works related to sound event annotation.

2.1 Asking for strong labels

Strong annotation allows annotators to freely choose class labels and segment boundaries (onset and offset) for the sound events. This can offer more detailed annotations, but demand more from the annotators.

In bioacoustics, expert annotators often use spectrograms to visualize the audio recording, and learn to recognize specific visual patterns of the sound events, helping them to annotate without even listening to the recordings. This technique enables them to annotate extensive audio data efficiently, focusing on specific classes of events like bird calls [14]. Strong labeling can be successful when using expert annotators. However, the need for experts means a higher cost, and also that the annotation may not scale, because there are only a limited number of experts.

A recent audio data collection effort for Anuran sound events (AnuraSet) tried to mitigate the problem of limited specialists by using first asking for weak labeling of 1 minute audio segments from both generalists (bioacoustic experts) and specialists (herpetologists), followed by asking for strong labeling of a subset of the audio segments only from the specialists [15].

Detecting the onsets and offsets of sound events is a hard task, and there is a lot of subjectivity in defining these boundaries for the events. For example, will a sequence of multiple footstep sound events be annotated as one long event or multiple short events? This depends on how the annotator interprets the task [16]. And detecting consistent boundaries for the sound event of a car passing by adds additional subjectivity to this [2], since the sound changes very gradually there are no clear boundaries. The fact that different annotators may interpret a strong labeling task in different ways is one of the reasons that weak labeling, a much simpler task, can be preferred. Especially, when we have no guarantee on the expertise of the annotators such as when crowdsourcing is applied [3].

As an example, a subset of 67K audio recordings from AudioSet has been strongly annotated. The onset and offset of events were given by multiple annotators, each annotator given the ability to correct the annotation from the previous annotator. However, even allowing 5 such passes this rarely resulted in a consensus [8].

Table 1: Large-scale audio datasets using variations of FIX labeling.

Dataset	Task	Fixed length
CHIME [17]	Single-pass multi-label	4 seconds
AudioSet [18]	Single-pass multi-label	10 seconds
MAESTRO Real [3]	Single-pass multi-label	10 seconds
OpenMIC-2018 [5]	Multi-pass binary-label	10 seconds

Hershey et al. [8] show that training even on a few strongly annotated recordings, in addition to weak labels, can result in better SED models than training only on weakly labeled data.

2.2 Asking for weak labels

A common way to simplify the annotation task is to segment the audio recording automatically, and then ask the annotator to perform the much easier task of assigning a class label to each audio segment. A common approach is FIX labeling.

Fixed audio segments

FIX weak labeling is a common annotation strategy. Some large-scale audio datasets using variations of FIX labeling are presented in table 1. Two different annotation tasks are usually considered: single-pass multi-label annotation and multi-pass binary-label annotation [6]. Single-pass multi-label means that annotators are asked to recognize the presence of multiple event classes during a single pass through the data, and multi-pass binary label means that we ask annotators to detect presence of a single event class at a time, and we perform one annotation pass for each class of interest. Cartwright et al. [6] has studied the trade-offs between multi-pass binary-labeling and single-pass multi-labeling and found that binary labeling is preferred when high recall is needed. They used 10 second segments.

The FIX labeling of AudioSet [18] was done for a selection of non-overlapping 10 second segments from the entire audio dataset. The information gained about temporal activations of sound events is therefore limited by the 10 second segments, that they do not overlap, and that they are not necessarily temporally adjacent. Further, Shah et al. [19] used AudioSet to simulate weak labeling using 10 (original), 30 and 60 second segments, and show that this leads to a drop in SED model performance.

In contrast, FIX labeling of MAESTRO Real [3] was done by crowdsourcing redundant weak labels for each 10 second segment, where segments overlap by 9 seconds. This procedure give a much more information about the temporal activation of sound events. The 9 second overlap means that we can get a temporal resolution of 1 second for isolated sound

events, but the 10 second segment also means that if two sound events occur closer than 10 seconds then the procedure will not label them as two distinct events.

In this work we restrict ourselves to study the properties of FIX labeling where the segments are adjacent, but do not overlap. Further we consider the setting of a multi-pass binary-label annotation task, which is suitable for crowdsourcing annotations because of the simplicity of the task. However, note that the theory derived for the multi-pass binary-label task can be applied also to the single-pass multi-label setting under certain assumptions on the data distribution and the annotator (see the discussion in section 8).

Adaptive audio segments

An alternative approach is to segment the audio recording using the structure of the sound events to better adapt each segment to the sound events of interest. The optimal method is ORC labeling (not known in practice) which use the ground truth segmentation of the audio. Adaptive audio segment methods try to model ORC labeling.

Zhao et al. [20] segment the audio recordings by using change point detection on the cosine distance between temporally neighbouring embeddings of the audio recording. The audio segments are pre-computed and then weakly labeled.

Martinsson et al. [13] propose an adaptive change point detection method which learns to better segment the audio with each new annotation during the annotation process.

Pseudo labeling or label propagation

Another approach, which is subtly different from adaptive audio segments, is to automatically adjust the weak labels to the data after they have been given. This is called pseudo labeling, or label propagation. The idea is that if we have a label for part of the data, we may be able to adjust the label using the structure of the data.

Shin et al. [21] propose a framework where weak labels are given to single points in time in a time series, the labels are then propagated to other temporally neighbouring data according to a temporal coherence criterion. For an audio recording this would mean that we simply have to click on a sound event when it appears to indicate presence, and then the presence label is automatically propagated to the rest of the event.

Dinkel et al. [22] propose to first train a model on weakly labeled audio data, then predict new pseudo labels for the same data, and then finally train a new model using the predicted labels.

There is a subtle difference between these approaches, where a weak label is first collected

and then adjusted, and methods using adaptive audio segments, where the segments are first adjusted and then weakly labeled. By first constructing the segments and then ask for weak labels we ensure that there is always a form of verification of the segmentation in the annotation process.

2.3 Our contribution

In this paper we model a version of FIX labeling to better understand the annotation quality that can be achieved for a given annotator model and sound event length distribution. We relate this to ORC labeling (upper bound not know in practice) to better understand the gap between these two weak labeling approaches. By quantifying the gap between FIX and ORC we can better understand the potential of adaptive audio segment methods and where current methods [20, 13] are placed within these limits.

In principle the analysis is not limited to audio, but applies in general to any time series with events under the stated assumptions.

3 Problem setting

We consider the multi-pass binary-labeling setting where we want to associate presence labels to different classes of sound events in audio recordings through multiple passes.

We model an audio distribution and an annotator which can only weakly label the presence or absence of sound events in a given audio segment. We are interested in the label quality of the presence annotations under different assumptions about the annotator model and the audio distribution.

Below we explain how the sound events for the audio data and the annotator are modeled in this work.

3.1 Audio data

A sound event consists of a start time $a_e \in \mathbb{R}$, end time $b_e \in \mathbb{R}$ and a class $c \in \mathcal{C}$, denoted as $e = (a_e, b_e, c)$ with event length $d_e = b_e - a_e$. We refer to the event timings (a_e, b_e) as the onset and offset, and c as the event class. The audio recordings are assumed to be of finite length T and we assume that the events e can occur everywhere with equal probability as long as the whole event is in the recording, i.e., $a_e \in [0, T - d_e]$.

3.2 Annotator model

For a given sound event class $c \in \mathcal{C}$, the annotator is modeled to indicate the presence or absence of that class of event in a given audio segment. Let $q = (a_q, b_q)$ denote the query segment where $a_q \in \mathbb{R}$ is the onset, and $b_q \in \mathbb{R}$ is the offset of the segment. The query length is $d_q = b_q - a_q$. We then simulate the process of asking an annotator if event class c is present in q. For example, using FIX labeling the audio recording is segmented into fixed length query segments, where the onset and offset of each query segment are illustrated as dashed red lines in figure 1, and the annotator is asked to indicate presence (red) or absence (white) of event class c for each segment.

An annotator will need to observe some part of the event to feasibly detect it's presence. We therefore model the annotator so that a fraction γ of the sound event e needs to be in the query segment q for the annotator to detect the presence of class c, which leads to definition 1 (event fraction) and definition 2 (annotator criterion) below.

Definition 1. The *event fraction* is the fraction of the total event present in a query segment,

$$h(e, q) = \frac{|e \cap q|}{d_e}, \tag{3.1}$$

where $e \cap q$ denotes the intersection between the query segment (a_q, b_q) and the event timings (a_e, b_e) of class c, and d_e the event length.

Definition 2. The *annotator criterion* γ is the smallest event fraction in query segment q assumed necessary for the annotator to detect the presence of event e,

$$h(e, q) \geq \gamma. \tag{3.2}$$

What this means is that for a given sound event class $c \in \mathcal{C}$ to annotate, if an annotator is given a query segment $q = (a_q, b_q)$ and it overlaps with a sufficient fraction γ of a sound event $e = (a_e, b_e, c)$ of the given class c, then the annotator will give the presence label (1) to q, and otherwise the annotator will give the absence label (0) to q.

A non-strict annotator criterion, when γ is close to 0, means that the simulated annotator can detect the presence of very small fractions of sound events in query segments, and a strict annotator criterion, when γ is close to 1, means that a large fraction of the sound event is needed for the annotator to detect it's presence in the query segment.

Defining the annotator criterion to be a fraction of the sound event makes sense for sounds where the meaning of the sound is only apparent after hearing enough of it. How large this fraction is depends on the meaning that we want to attribute to the sound.

Let us consider an audio recording of a human who speaks a complete sentence. To give the presence label indicating "a human is talking" the annotator can probably listen to any

x seconds of the recording. However, if we want to attribute a class label that conveys more meaning, for example, "a human is talking about an animal", then the annotator will need to listen to enough of the spoken sentence to deduce that meaning.

Another example is for detecting the presence of bird song in bioacoustics. Bird song is made up of a sequence of notes, and the meaning that is conveyed depends on this sequence. To understand the class of the the bird song the annotator will often need to hear at least a fraction of these notes.

The annotator criterion is an assumption about this fraction. When $\gamma = 1$ we say that the annotator would need to hear the whole sentence spoken by the human or the whole sequence of notes sung by the bird in the above examples to detect the presence of the meaning that we seek. And conversely, when γ is close to 0 we say that the annotator need to hear almost none of it.

The only way to be sure that the annotator can detect the presence of the event is when $\gamma = 1$, all other choices for γ are assumptions about the annotators ability to detect the events, which in practice will depend on both the annotator and the type of event.

3.3 Quality of the presence labels

We do not model the type of annotation error where an annotator can falsely detect the presence of sound events or falsely miss the presence of a sound event. Instead, we focus on the type of annotation error introduced by the imprecise onsets and offsets which is typical for weak labeling.

Note that when a given query segment q does not overlap with any sound event e of class c, i.e., $h(e, q) = 0$, then the annotator will always indicate absence. That is, in the non-overlapping case the absence annotation will always be correct. We therefore do not include these cases when formulating the QIoU metric in the next section. Instead we focus only on the the presence labels, which can only occur in the case of overlap, i.e, $h(e, q) > 0$. A reason for this is that the absence class is typically a large majority and including it would mean that the more absence sounds in the data the better the overall score, which is a property that we do not want for the metric.

We are interested in the annotation quality of the presence labels in expectation for different annotator models and data distributions when using FIX labeling and ORC labeling.

4 Analysis of FIX

In this section we define the FIX labeling method and derive an expression for the expected QIoU of the event labels. We assume a simplified sound event distribution where audio recordings only contain a single event with a deterministic event length, arguably the simplest sound event distribution to annotate. In section 7.1 we use this expression, derived for the simplified case, to study more complex distributions with varying event lengths and varying number of events.

4.1 The FIX labeling method

The FIX method splits the audio recording into equal length query segments and then ask an annotator to give each of these a class label. This method is commonly used in practice [10, 11, 12], which is why we want to understand the limits better. Let B_{FIX} denote the number of query segments that we are allowed to use, then the query segments are defined as

$$\mathbb{Q}_{\text{FIX}} = \{(a_0, b_0), (a_1, b_1), \ldots, (a_{B_{\text{FIX}}-1}, b_{B_{\text{FIX}}-1})\} = \{q_0, \ldots, q_{B_{\text{FIX}}-1}\} \tag{4.1}$$

where the start and end timings of each query segment is $q_i = (a_i, b_i) = (id_q, (i+1)d_q)$ and the fixed query segment length is $d_q = T/B_{\text{FIX}}$.

4.2 The query intersection over union (QIoU)

Definition 3. The query intersection over union (*QIoU*) is defined as the overlap between the query q and the event e divided by the query length d_q when the annotator criterion is fulfilled, and otherwise 0,

$$F(e, q, \gamma) = \begin{cases} \frac{|e \cap q|}{d_q}, & \text{if } h(e, q) \geq \gamma, \\ 0, & \text{if } h(e, q) < \gamma. \end{cases} \tag{4.2}$$

QIoU is designed to only measure the annotation quality of the presence class, not the absence class. The considered annotator model can not falsely detect the presence of an event in a query segment with no overlap with an actual event. Therefore, for all query segments that do not overlap with an event we have perfect absence labels. Presence annotations can only be given by the annotator model when there is an overlap between the given query segment q and an event e of class c. We therefore only consider the cases when there is overlap (see also section 3.3).

If the annotator criterion is fulfilled, then the annotator detects the presence of a sound event e of class c in query q, for the considered event class $c \in \mathcal{C}$. In this case $F(e, q, \gamma) = \frac{|e \cap q|}{d_q}$, which is equivalent to the event density for query q [19, 23]. Event density is the fraction of the query q that contains the event, that is, the fraction of true positives. Conversely, $1 - F(e, q, \gamma)$ is the fraction of false positives. A QIoU of 1 therefore means that the query segment weakly labeled with the presence class only contains the presence class, and thus has an event density of 1.

If the annotator criterion is not fulfilled, then this means that we entirely miss the fraction of the event that overlaps with the query, and we set the QIoU to $F(e, q, \gamma) = 0$.

Note, that a high QIoU for the presence class will necessarily result in high quality absence labels. When the average QIoU over all query segments is 1 this means that we have perfect labels for both the presence and absence classes.

4.3 The expected QIoU for FIX

In this section we derive an expression for the QIoU in expectation over a distribution of audio recordings with sound events to annotate. Since the annotator criterion is never fulfilled when there is no overlap between a query and an event and we therefore always get the absence label, we know that the absence labels are perfect in this case. The only case where the labels are not necessarily perfect is when there is overlap. We therefore consider only the cases with overlap and derive the QIoU in expectation for the presence class and a given sound event distribution. Let $e_t = (a_q - d_e + t, a_q + t)$ denote the event that ends at the start of the query segment when $t = 0$, and that starts where the query segment ends when $t = d_e + d_q$. Now, $t \in [0, d_e + d_t]$ describes all possible occurrences of overlap.

To get the expected QIoU we therefore now need to compute

$$\mathbb{E}_{\mathrm{t} \sim p}\left[F(e_{\mathrm{t}}, q, \gamma)\right] = \int_0^{d_e + d_q} F(e_t, q, \gamma) p(t) \mathrm{d}t, \tag{4.3}$$

$$= \frac{1}{d_e + d_q} \int_0^{d_e + d_q} F(e_t, q, \gamma) \mathrm{d}t. \tag{4.4}$$

where $p(\mathrm{t})$ is the probability distribution for the realizations of t. Since we assume that the sound event can occur anywhere in the audio recording with equal probability we get $p(\mathrm{t}) = 1/(d_e + d_q)$.

We plot $F(e_t, q, \gamma)$ for $t \in [0, d_e + d_q]$ in figure 2, and see that there are five distinct areas under the curve, some of them symmetrical around the center. Computing the integral of $F(e_t, q, \gamma)$ for $t \in [0, d_e + d_q]$ leads to theorem 1.

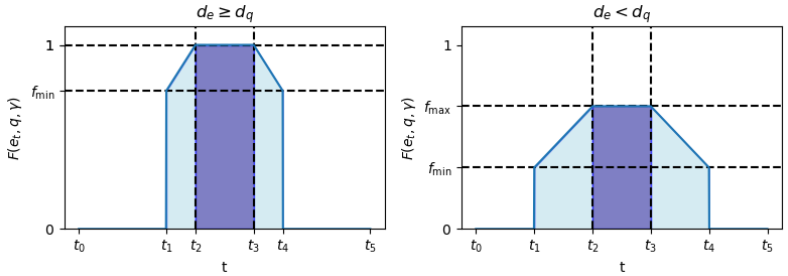


Figure 2: The QIoU $F(e_t, q, \gamma)$ for $t \in [t_0, t_5]$, where $t_0 = 0$ and $t_5 = d_e + d_q$. In appendix 1 we show how t_1, t_2, t_3 and t_4 can be defined as functions of d_e, d_q and γ. Here, they are abstract and indicate the states t where $F(e_t, q, \gamma)$ changes behaviour.

Theorem 1. The expected QIoU of an arbitrary query segment q of length d_q which overlaps with an event e of length d_e using annotator criterion γ is

$$P(d_e, d_q, \gamma) = \mathbb{E}_{t\sim p}\left[F(e_t, q, \gamma)\right] = \begin{cases} \frac{1}{d_e + d_q}\left[d_e - \frac{d_e^2\gamma^2}{d_q}\right], & \text{if } d_q \geq \gamma d_e, \\ 0, & \text{if } d_q < \gamma d_e. \end{cases} \tag{4.5}$$

Proof. See appendix 1.1 for the full proof. Basically, the plot of $F(e_t, q, \gamma)$ for $t \in [0, d_e + d_q]$ in figure 2 show that there are five different regions to integrate over: (t_0, t_1), (t_1, t_2), …, (t_4, t_5). We break the integral in Eq. 4.4 into five separate integrals, and we show how to define $t_0, \ldots, t_5$ in terms of d_e, d_q and γ and then solve each of the integrals separately for the cases: $d_e \geq d_q$ and $d_e < d_q$. In both cases we arrive at the expression $\frac{1}{d_e+d_q}\left[d_e - \frac{d_e^2\gamma^2}{d_q}\right]$ for the expected QIoU, and finally taking the case where $d_q < \gamma d_e$, when the annotator criterion can never be fulfilled and $F(e_t, q, \gamma)$ is always 0, into consideration we arrive at theorem 1. □

Note that in theorem 1 we have implicitly assumed that only one sound event overlaps with any given query segment. This is not always true in practice, but we show by simulation in section 7.1 that breaking this assumption does not have a large effect.

Theorem 1 gives us an expression for the expected QIoU when query segments of length d_q are used to detect events of deterministic length d_e for a given annotator criterion γ. We use theorem to prove the next theorem, which states what query segment length d_q that should be used to maximize the expected QIoU.

Theorem 2. The query length that maximize the expected QIoU is

$$Q_{\max}(d_e, \gamma) = \sqrt{d_e^2(\gamma^2 + \gamma^4)} + d_e\gamma^2. \tag{4.6}$$

Proof.

$$\frac{\partial P}{\partial d_q} = \frac{d_e(d_e^2\gamma^2 + 2d_e d_q\gamma^2 - d_q^2)}{d_q^2(d_e + d_q)^2}, \tag{4.7}$$

and solving $\frac{\partial P}{\partial d_q} = 0$ for d_q under the constraint that $d_q \geq \gamma d_e$, $d_e > 0$, and $\gamma \geq 0$ gives the solution $Q_{\max}(d_e, \gamma) = \sqrt{d_e^2(\gamma^2 + \gamma^4)} + d_e\gamma^2$. We have visually checked that the second partial derivative $\frac{\partial^2 P}{\partial d_q^2}$ is negative at $d_q^{(\max)}$ for $\gamma \in (0, 1]$. Which makes this a local maxima. □

Theorem 2 tells us which query length d_q that should be used for FIX labeling to maximize the expected QIoU. We can use this expression to better understand the annotation cost (number of query segments) necessary to maximize QIoU in expectation. And by inserting setting $d_q = Q_{\max}(d_e, \gamma)$ in theorem 1 we get an expression for the maximum expected QIoU.

Theorem 3. The maximum expected QIoU for an arbitrary event of length d_e is

$$P_{\max}(\gamma) = 1 + 2\gamma^2 - 2\sqrt{\gamma^2 + \gamma^4}. \tag{4.8}$$

Proof. From theorem 2 we know that $Q_{\max}(d_e, \gamma) = \sqrt{d_e^2(\gamma^2 + \gamma^4)} + d_e\gamma^2$ maximize the expected QIoU $P(d_e, d_q, \gamma)$. We can thus define the max QIoU

$$\begin{aligned} P_{\max}(\gamma) &= P(d_e, Q_{\max}(d_e, \gamma), \gamma) && (4.9)\\ &= \frac{1}{d_e + \sqrt{d_e^2(\gamma^2 + \gamma^4)} + d_e\gamma^2}\left[d_e - \frac{d_e^2\gamma^2}{\sqrt{d_e^2(\gamma^2 + \gamma^4)} + d_e\gamma^2}\right] && (4.10)\\ &= 1 + 2\gamma^2 - 2\sqrt{\gamma^2 + \gamma^4} && (4.11) \end{aligned}$$

□

Note that $P_{\max}(\gamma)$ is only a function of γ. That is, the maximum expected performance is independent of the event length. Intuitively we only need a longer queries if we have longer events.

Finally, we use the expression for the query segment length $Q_{\max}(d_e, \gamma)$ that maximize the expected QIoU to compute how many query segments are needed for an audio recording of length T.

Theorem 4. The number of queries $B_{\text{FIX}}^{(\max)}$ which are needed by FIX to maximize the QIoU in expectation for an audio recording of length T when $d_e = 1$ is

$$B_{\text{FIX}}^{(\max)} = \frac{T}{\gamma(\gamma + \sqrt{1 + \gamma^2})}. \tag{4.12}$$

Proof. If you divide a recording of length T into $B_{\text{FIX}}^{(\max)}$ equal length segments you get a segment length of $\gamma(\gamma + \sqrt{1+\gamma^2})$, which by theorem 2 maximizes the expected QIoU for $d_e = 1$. □

Theorem 4 tells us the number of query segments needed to maximize the expected QIoU when FIX labeling audio recordings of length T with a single randomly occurring event of length d_e. Since there is arguably no simpler audio data distribution to annotate than that where recordings only contain single events of a deterministic length (except for when no events occur at all). We can treat $P_{\max}(\gamma)$ as an upper bound on the maximum expected QIoU for any audio distribution (we see this empirically in the results in section 7), and $B_{\text{FIX}}^{(\max)}$ is tells us how many segments the annotator has to weakly label.

5 Simulation of FIX labeling

To validate the theory derived in the previous section, we simulate FIX labeling of different audio recording distributions and compare the average simulated label quality with theoretical results from section 7.

We sample 1000 audio recordings of length T with M events. Each event length is sampled from either a uniform, normal, or gamma distributed event length distribution. We also consider the sample of a real event length distribution for dog barks and baby cries from the NIGENS dataset [24]. An event of the sampled length is then randomly placed in the recording, where the start time a_e is drawn uniformly at random from $[0, T - d_e]$.

Since we only consider the absence or presence of an event class, we merge overlapping events. This means that for when many events are sampled (large M), the event length distribution can change due to the merging. Neither FIX nor ORC can handle overlapping events without additional information from the annotator. The optimal way to resolve these using an annotator model that can answer, e.g., how many events are occurring in a given query segment is an orthogonal but interesting research direction.

We then compute the average QIoU over the sampled audio recordings for different annotator criterion γ and query lengths d_q. We do this for $\gamma \in [0.01, 0.99]$, and $d_q \in [d_{e,\min}/100, 2.5d_{e,\max}]$. The range of d_q is set because we can not feasibly search over all possible d_q. In theory, for $\gamma = 1$ the query length that maximize the expected QIoU is $d_e(1+\sqrt{2}) \approx 2.5d_e$, and for $\gamma = 0.01$ it is $d_e(\gamma(\gamma + \sqrt{1+\gamma^2})) \approx d_e/100$. We therefore compute the minimum sampled event length $d_{e,min}$ and the maximum sampled event length $d_{e,max}$ from the 1000 audio recording samples, and set the limit according to this and the theory. We can then get the maximum average QIoU for each considered γ and the corresponding optimal query length to compare with the theory.

6 Analysis of ORC

The ORC labeling method uses the ground truth event labels to construct the query segments, and then ask the annotator to give a class label to these. The query segments are constructed using the true event labels

$$\mathbb{Q}_{\text{ORC}} = \{(a_0, b_0), (a_1, b_1), \ldots, (a_{B_{\text{ORC}}-1}, b_{B_{\text{ORC}}-1})\}, \tag{6.1}$$

where (a_i, b_i) is the ith true event label. B_{ORC} is the sufficient number of queries to get a QIoU of 1. The number of queries needed grow linearly with the number of target events M in the given audio recording.

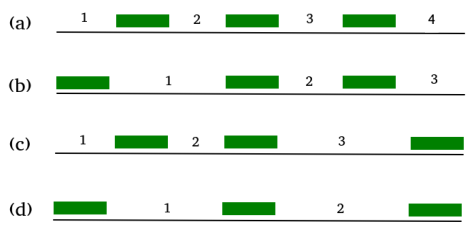


Figure 3: The four cases with different number of absence segments (numbered) between the events (green) in an audio recording. In this example we have 3 events, and either (a) 4, (b) 3, (c) 3, or (d) 2 absence segments. On average 3 absence segments if assumed equally probable.

In figure 3 we show an example with $M = 3$ sound events without overlap. Between each sound event there is a segment with the absence of an event giving $M-1$ absence segments in total (see (d) in figure 3) then we potentially have 1 extra absence segment at each end of the recording (see (b-d) in figure 3). If we assume these cases to be equally probable we end up with an average of M absence segments. The total number of segments needed on average to query the audio recording perfectly and get a QIoU of 1 is therefore

$$B_{\text{ORC}} = 2M. \tag{6.2}$$

This tells us how few queries we can use and still get perfect labels with ORC. This is used as a reference and an upper bound on what is achievable.

7 Results

In this section we present the results of the simulated annotation process, and show how these connect to the derived theory. We start by looking at the annotation quality for FIX and ORC, and then at the annotation cost.

7.1 Annotation quality

Single event with deterministic length

These results are derived using the simulation setup described in section 5, with $M = 1$ (a single event) and $d_e = 1$ (deterministic length).

In figure 4 we show the simulated maximum expected QIoU (left) and the corresponding query length for different γ (right). $P_{\max}(\gamma)$ is the maximum expected QIoU achievable with annotator criterion γ for the considered event length distribution. We can see that the simulated average QIoU closely follows the expected QIoU, and that the corresponding segment length leading to this maximum is the same in in theory and simulation.

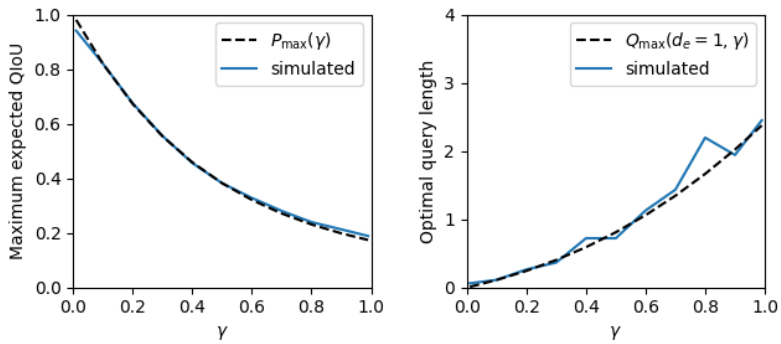


Figure 4: In the left panel we show the maximum expected QIoU, $P_{\max}(\gamma)$, for different γ, and the average maximum QIoU from the simulations. In the right panel we show the query length that leads to this maximum in theory, $Q_{\max}(d_e, \gamma)$, and in simulation. The theory follows the simulations well.

Figure 4 tells us that, for example, if the annotator needs to hear more than 50% of the sound event to detect presence ($\gamma = 0.5$) then the best label quality that can be achieved is $P_{\max}(0.5) \approx 0.38$. This means that on average around 62% of the annotated query segments contain absence sound (a lot of false positive labels). We also see that the query length that give the maximum QIoU is $Q_{\max}(1, 0.5) \approx 0.96$. There is a large gap to the ORC method which always give a QIoU of 1 as long as at least $B_{\text{ORC}} \geq 2M$ queries are used. In general, we can see how the maximum QIoU deteriorates with a growing γ, and which query segment length to choose to maximize QIoU in expectation.

The simulations validate that the theory holds for the case where audio recordings contain a single event of deterministic event lenght ($d_e = 1$).

Single event with stochastic length

In the previous simulations and for the derived theory we have assumed a single event with deterministic length, now we relax this assumption to stochastic event lengths. We do this to study the effect of the event length distribution on the maximum expected QIoU and the optimal query length. The expected QIoU over a distribution of event lengths for a given γ and query segment length d_q can be written as

$$P(d_q, \gamma) = \mathbb{E}_{\mathrm{d}_e \sim p}\left[P(d_e, d_q, \gamma)\right] \tag{7.1}$$

$$= \int_0^\infty P(d_e, d_q, \gamma) p(d_e) \mathrm{d}d_e, \tag{7.2}$$

where $p(d_e)$ denotes the probability distribution for the outcomes of the random event length variable $\mathrm{d}_e \sim p(d_e)$.

In each figure we present the derived theoretical rules for the simplified event length distribution $P_{\max}(\gamma)$ and $Q_{\max}(d_e, \gamma)$, the results from integration of Eq. 7.2 with different event length distributions $p(d_e)$ (numerical), and the simulated results using the procedure described in section 5 (simulated).

Since $Q_{\max}(d_e, \gamma)$ is derived for a deterministic event length d_e, and require a choice of this value, we set d_e to the average event length for each distribution in these experiments as a heuristic. We then present the maximum expected QIoU for different γ (left in figures) and the query segment length that maximize the expected QIoU (middle in figures), and the histogram for the considered event length distributions (right in figures).

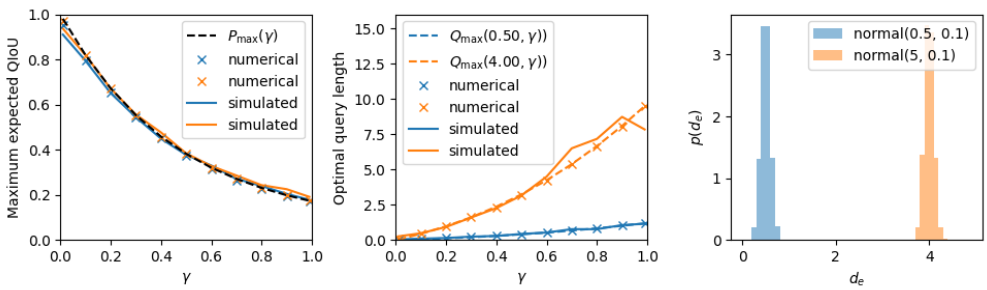


Figure 5: Normal distributions with different means.

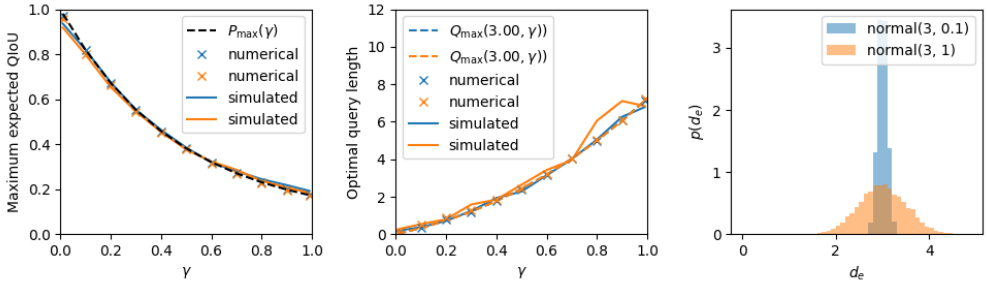


Figure 6: Normal distributions with different variances.

In figure 5 and figure 6 we see that the mean and variance of the normal distribution has a small (if any) effect on the maximum expected QIoU, but the mean does affect which query segment length that maximize the expected QIoU. We also see that $Q_{\max}(\mu, \gamma)$ follow the simulated and numerical optimal query length well for all considered normal distributions, when μ is set to the average event length for the considered event length distribution. The average event length can be used as a heuristic value if we only know the average and not the true distribution to integrate over.

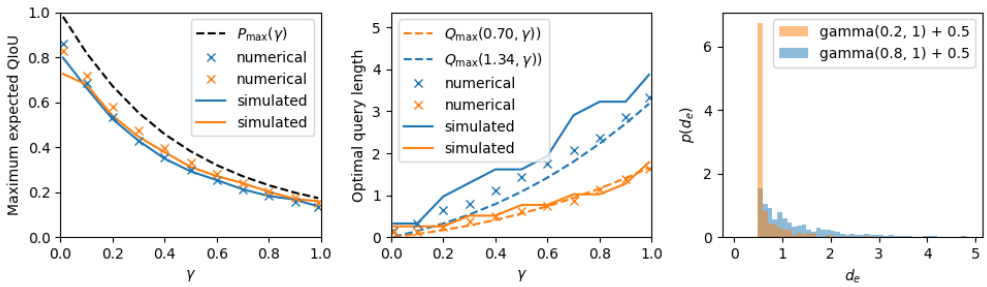


Figure 7: Gamma distributions with different shape and scale parameters.

In figure 7 we can see that a gamma distribution does affect the maximum expected QIoU, and that simply setting d_e to the average event length of the distribution in $Q_{\max}(d_e, \gamma)$ leads to underestimating the optimal query length. The gamma event length distribution hard for FIX, since FIX labeling can not achieve good label quality for short events and long events at the same time. The gamma distributions considered here have a high probability of short events, and decreasing probability for longer events, which means that there is a trade-off for the FIX labeling method which must optimizing for the many short events (and possibly miss the long events due to the annotator criterion), or optimize for the long events (and thus necessarily give imprecise onset and offsets for the short events).

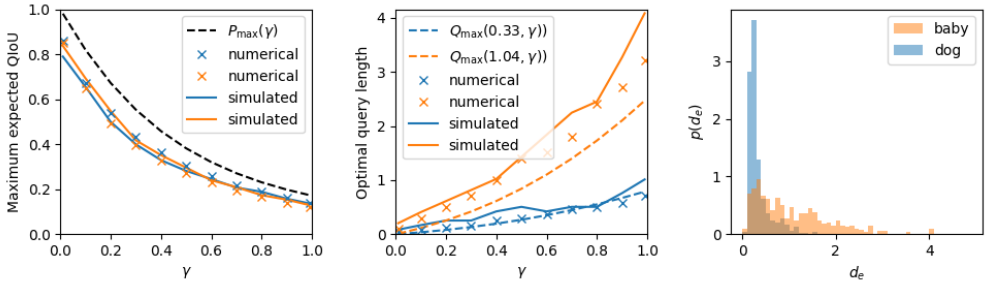


Figure 8: Dog and baby event length distributions from the NIGENS dataset [24]. These annotations have been made with a strong guarantee for high quality onsets and offsets.

In figure 8 we validate the theory against a real sample of event lengths from either baby cries or dog barks. Numerical integration between the derived expression and the histogram predicts the simulations well.

Multiple events with stochastic length

In these experiments we allow multiple events to occur in the same recording ($M > 1$), and therefore no longer enforce the implicit assumption that a query segment never overlaps with more than one event.

In figure 9 we sample 30 events of length $d_e = 1$ for each audio recording. We can see that this does have a an effect on the expected maximum QIoU and the corresponding query length, but not that large.

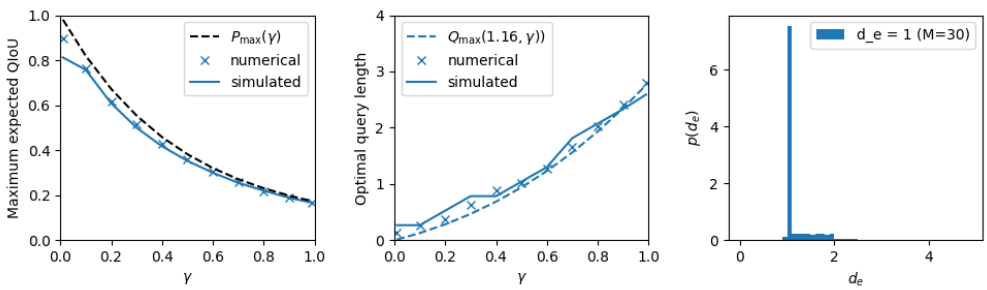


Figure 9: 30 events with event length $d_e = 1$ occur in each sampled audio recording.

In figure 10 we sample 100 events of length $d_e = 1$ for each audio recording. This is an extreme case, where the event density of the recording is very high. We would not expect this to happen in practice. But, now the effect starts to show. However, the theory still

matches the simulations.

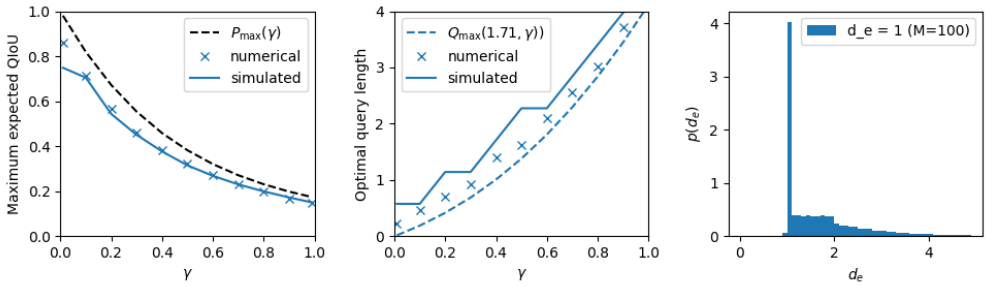


Figure 10: 100 events with event length $d_e = 1$ occur in each sampled audio recording.

In all experiments we can see that $P_{\max}(\gamma)$ acts as an upper bound on the expected QIoU for any of the other considered event length distributions. Arguably, any recording distribution which has more than a single event, or events of varying length should be at least as hard to annotate with FIX as the one with only a single event of deterministic event length.

7.2 Annotation cost

Since we now know the optimal query length, we can study the cost of getting maximum expected QIoU for different annotator models. Let us consider the cost model

$$C(T, B) = (1 - r)T + rB \tag{7.3}$$

where $1 - r$ is the cost of listening to a second of audio, and r is the cost of answering a query. We can then look at the relative cost of annotating an audio recording of length T with M sound events of length $d_e = 1$ using either FIX or ORC. The cost is computed for the number of queries that maximize the expected QIoU for both methods. For FIX this is $B_{\text{FIX}}^{(\max)} = T/\gamma(\gamma + \sqrt{1 + \gamma^2})$ (see theorem 4). We study the cost of FIX for this optimal query length, and compare with different costs for ORC. For ORC the expected QIoU is 1 as long as $B_{\text{ORC}} \geq 2M$ queries are used. In practice we need to estimate B_{ORC}. We simulate different degrees of overestimation by $B_{\text{ORC}} = s2M$ for $s \in \{1, 2, 4, 8\}$. That is, when $s = 1$ exactly the necessary number of queries are used and when $s > 1$ more queries are used than actually needed.

The estimate can in practice be independent of the audio recording and set as a bound on M. E.g., if we never expect to encounter more than $M_{\max}$ sound events in a recording of length T we set $B_{\text{ORC}} = 2M_{\max}$. But, it could also depend on the audio recording and be an estimate of the number of events that occur. In this analysis we assume that it is unlikely to overestimate the number of events by more than a factor of 8.

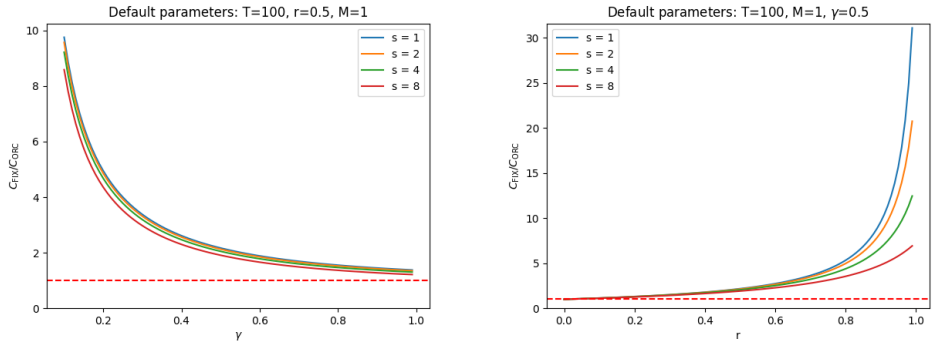


Figure 11: The relative cost of FIX and ORC for varying: annotator criteria γ (left), and cost ratios r (right). The default parameters are: $T = 100$, $r = 0.5$, $M = 1$ and $\gamma = 0.5$. We simulate overestimating the number of needed queries $B_{\text{ORC}} = s2M$ by a factor of s for $s \in \{1, 2, 4, 8\}$ to see how this affect the relative cost. The cost of FIX is greater than the cost of ORC above the dashed red line where the cost ratio is 1.

The relative cost we study becomes

$$\frac{C_{\text{FIX}}}{C_{\text{ORC}}} = \frac{C(T, B_{\text{FIX}}^{(\max)})}{C(T, B_{\text{ORC}})}. \tag{7.4}$$

In figure 11 we plot the relative cost for different: annotator criteria ($\gamma \in [0.1, 1]$), and cost models ($r \in [0, 1]$) and in figure 12 for different number of sound events ($M \in \{1, 2, \ldots, 100\}$), and audio recordings lengths ($T \in [1, 100]$). We change one of these parameters at a time, and the default values are: $T = 100$, $r = 0.5$, $\gamma = 0.5$, and $M = 1$. The default setting simulates the cost of annotating a 100 second audio recording ($T = 100$), where the cost of listening to a second of audio and handling a query is the same ($r = 0.5$), the annotator need to hear at least 50% of the sound event to detect presence ($\gamma = 0.5$), and there is one sound event in the recording ($M = 1$).

We know from the previous section that the only way to achieve an expected QIoU of 1 with FIX is when $\gamma \to 0$, then $B_{\text{FIX}}^{(\max)} \to \infty$, and $C_{\text{FIX}}/C_{\text{ORC}} \to \infty$. That is, we need to pay an infinite cost to achieve an expected QIoU of 1 with FIX, but only a finite cost with ORC.

We see this in the left panel of figure 11, where the ratio increases drastically as $\gamma \to 0.1$. We limit γ to 0.1 because otherwise we get so large values that the lines become indistinguishable. We can see that in the default setting, it does not matter what γ we choose for the annotator, the cost of the ORC method is strictly less than the cost of FIX. But for large γ the cost becomes more similar. What this tells us in combination with theorem 3 is that FIX can give similar expected QIoU as ORC but at a much higher cost, and that it can have as low cost as ORC but with a much lower expected QIoU.

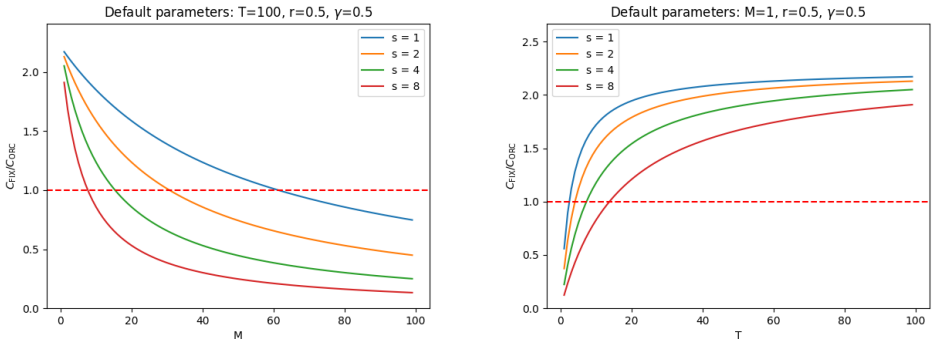


Figure 12: The relative cost of FIX and ORC for varying: number of sound events M (left) and recording lengths T (right). The default parameters are: $T = 100$, $r = 0.5$, $M = 1$ and $\gamma = 0.5$. We simulate overestimating the number of needed queries $B_{\text{ORC}} = s2M$ by a factor of s for $s \in \{1, 2, 4, 8\}$ to see how this affects the relative cost. The cost of FIX is greater than the cost of ORC above the dashed red line where the cost ratio is 1.

In practice a γ that is close to 0 is not only infeasible because of the large associated cost, it is also not a realistic assumption about the annotator. This would require the annotator to be able to detect event presence at the smallest possible time scale, i.e., from an audio sample alone.

In the right panel of figure 11 we see that it does not matter which r we choose for the cost model, the cost of ORC is strictly less than the cost of FIX in the default setting for different r.

In the left panel of figure 12 we can see that the number of events in the recording do affect the cost ratio. When $s = 1$ we can see that when no more than 60 events are in the audio recording then ORC is less costly than FIX. In the other extreme, when $s = 8$, FIX is less costly when at least roughly 10 events are present. That is, the number of events that occur in the recording determine when the ORC method is more costly than FIX.

In the right panel of figure 12 we observe a similar thing. But now the length of the recording changes instead of the number of events. When the recording is short (high event density) we no longer see a cost benefit of using ORC.

However, remember that the highest achievable expected QIoU when $\gamma = 0.5$ (default setting in these experiments) is $P_{\max}(0.5) \approx 0.382$, and here we compare to ORC which will have an expected QIoU of 1.0, so the additional cost incurred by ORC may be merited in some of these settings because of the much higher label quality.

8 Discussion

Some form of FIX labeling has been employed in many works. Theorem 2 can be used as a rule of thumb to choose the best segmentation length for a given event length. The results suggest that in most cases knowing the average event length is enough to get a sufficiently good estimate, but knowing the approximate distribution of event lengths is even better.

Extending the theory to more dimensions. In this paper we have only derived this theory for annotation of data measured along 1 dimension, i.e., sound event annotation. However, with slight adjustments a more general theoretical framework covering up to 3 dimensions should be feasible to derive, which would also cover FIX labeling of rectangles in e.g., images or FIX labeling of cubes in e.g, point clouds.

Multiple different presence classes. If there are m presence classes to annotate and the same annotator criterion γ holds for each of them then theorem 1 is applicable for the joint event length distribution of the different classes. Simply perform the same numerical integration with the joint event length distribution (if it can be estimated) as done in section 7.1. Of course, in reality different annotation criteria γ can be expected for different event classes (and in general different models entirely), in which case a more elaborate framework needs to be developed. Empirical studies that try to understand the properties of real annotators to understand how to best model them would be very beneficial.

$P_{\max(\gamma)}$ can be considered as an upper bound. The expression for the expected QIoU is derived for, arguably, the simplest sound event distribution to annotate, when there is only one event with a deterministic event length present. We can see in all of the results that $P_{\max(\gamma)}$ is higher than or equal to all the simulated (and numerically integrated) average QIoU for all of the more complex sound event distributions considered. Supporting the claim that this can be treated as an upper bound. A proof that shows that adding more events or introducing event length variability can only lead to a harder distribution to annotate has however not been given in this paper.

Better understand the properties of a SED model trained with FIX labels. We can use theorem 3 to understand the properties of the best performing model when the evaluation set contains FIX labels. For example, we now know that for $\gamma = 0.5$ the annotations will at most have an expected QIoU of $P_{\max}(0.5) \approx 0.38$ When evaluating a model with these labels the best performing SED model will therefore not be the one that solves the task perfectly, but a model that predict the onsets and offsets that match that of FIX labeling, which we now know would contain a large fraction of absence sound. We may end up rejecting the model that does what we really want simply because our evaluation criteria is insufficient. That is, if our objective is a model that does not have these properties we may have to consider other annotators (smaller γ) at an increased cost, or other ways of annotating at least the evaluation data.

Better understand properties of segment based evaluation criterias. The theory can be relevant for some evaluation procedures of sound event detection methods. A common way to evaluate models is to use a segment-based F_1 score [25], which in principle is similar to the FIX labeling. The audio recording is divided into fixed length segments, and the segments are given a label according to the overlap with ground truth labels. A difference is that since in evaluation we know the ground truth we basically have $\gamma \to 0$. Therefore, the expected QIoU becomes $P(d_e, d_q, 0) = d_e/(d_e + d_q)$, which tells us that the metric will accept a fraction of $1 - d_e/(d_e + d_q)$ absence sound in the presence annotations and still consider them as positives. Of course choosing the segment length $d_q \to 0$ will make $1 - d_e/(d_e + d_q) \to 0$. However, even for evaluation this choice is associated with an cost. The theory may be used to make sure that the accepted fraction of absence sound in the presence labels is satisfactory for the application at hand.

The annotator criterion γ depends on the interface for the annotator. Let us assume that the annotator has an interface which displays other information such as the spectrogram of the sound recording, and which lets the annotator click on the query segments to label presence of an event (similar to figure 1). There is almost no cost associated with looking at a spectrogram of a sound recording, if it is reasonably small. If the annotator can detect event presence based on the spectrogram, this could effectively be considered as a very small γ. The length of the query segments may not really matter for detecting presence in this setting, because the presence is detected at a recording level. But, they do still affect the resulting QIoU. But the cost of $d_q \to 0$ would be infinitely many clicks since the annotator still need to click on all the segments that contain the event. While the annotator criterion may be less applicable in these cases where presence is detected on a recording level, the theory still give a bound on the expected QIoU and the cost associated with that.

9 Conclusions

The results show a large gap between FIX and ORC in both the cost of annotation and the resulting expected label quality, a gap that could be reduced by developing annotation methods that better model ORC.

References

[1] Annamaria Mesaros, Toni Heittola, Tuomas Virtanen, and Mark D. Plumbley. Sound Event Detection: A tutorial. *IEEE Signal Processing Magazine*, 38(5):67–83, 2021. ISSN 15580792. doi: 10.1109/MSP.2021.3090678. URL `http://arxiv.org/abs/2107.05463{%}0Ahttp://dx.doi.org/10.1109/MSP.2021.3090678`.

[2] Annamaria Mesaros, Toni Heittola, and Dan Ellis. Datasets and evaluation. In Tuomas Virtanen, Mark Plumbley, and Dan Ellis, editors, *Computational Analysis of Sound Scenes and Events*, chapter 6, pages 147–179. Springer Cham, 2017.

[3] Irene Martin-Morato and Annamaria Mesaros. Strong Labeling of Sound Events Using Crowdsourced Weak Labels and Annotator Competence Estimation. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 31:902–914, 2023. ISSN 23299304. doi: 10.1109/TASLP.2022.3233468.

[4] Irene Martin-Morato, Manu Harju, and Annamaria Mesaros. Crowdsourcing Strong Labels for Sound Event Detection. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 246–250, 2021. ISSN 19471629. doi: 10.1109/WASPAA52581.2021.9632761.

[5] Eric J. Humphrey, Simon Durand, and Brian McFee. OpenMIC-2018: An open dataset for multiple instrument recognition. *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018*, pages 438–444, 2018.

[6] Mark Cartwright, Graham Dove, Ana Elisa Méndez Méndez, Juan P. Bello, and Oded Nov. Crowdsourcing Multi-label Audio Annotation Tasks with Citizen Scientists. *Conference on Human Factors in Computing Systems - Proceedings*, pages 1–11, 2019. doi: 10.1145/3290605.3300522.

[7] Qiuqiang Kong, Yong Xu, Wenwu Wang, and Mark D. Plumbley. Sound event detection of weakly labelled data with cnn-transformer and automatic threshold optimization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28: 2450–2460, 2020. doi: 10.1109/TASLP.2020.3014737.

[8] Shawn Hershey, Daniel P.W. Ellis, Eduardo Fonseca, Aren Jansen, Caroline Liu, R. Channing Moore, and Manoj Plakal. The benefit of temporally-strong labels in audio event classification. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 366–370, 2021. ISSN 15206149. doi: 10.1109/ICASSP39728.2021.9414579.

[9] Tiago A. Marques, Len Thomas, Stephen W. Martin, David K. Mellinger, Jessica A. Ward, David J. Moretti, Danielle Harris, and Peter L. Tyack. Estimating animal population density using passive acoustics. *Biological Reviews*, 88(2):287–309, 2013. doi: https://doi.org/10.1111/brv.12001. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/brv.12001`.

[10] Shuyang Zhao, Toni Heittola, and Tuomas Virtanen. Active learning for sound event classification by clustering unlabeled data. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 751–755, 2017. ISSN 15206149. doi: 10.1109/ICASSP.2017.7952256.

[11] Shuyang Zhao, Toni Heittola, and Tuomas Virtanen. An active learning method using clustering and committee-based sample selection for sound event classification. *16th International Workshop on Acoustic Signal Enhancement, IWAENC 2018 - Proceedings*, pages 116–120, 2018. doi: 10.1109/IWAENC.2018.8521336.

[12] Yu Wang, Mark Cartwright, and Juan Pablo Bello. Active Few-Shot Learning for Sound Event Detection. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 1551–1555, 2022. ISSN 19909772. doi: 10.21437/Interspeech.2022-10907.

[13] John Martinsson, Olof Mogren, Maria Sandsten, and Tuomas Virtanen. From Weak to Strong Sound Event Labels using Adaptive Change-Point Detection and Active Learning. In *EUSIPCO 2024 - 32nd European Signal Processing Conference*, 2024. URL `http://arxiv.org/abs/2403.08525`.

[14] Mark Cartwright, Ayanna Seals, Justin Salamon, Alex Williams, Stefanie Mikloska, Duncan MacConnell, Edith Law, Juan P. Bello, and Oded Nov. Seeing sound: Investigating the effects of visualizations and complexity on crowdsourced audio annotations. *Proceedings of the ACM on Human-Computer Interaction*, 1(CSCW):–246, 2017. ISSN 25730142. doi: 10.1145/3134664.

[15] Juan Sebastián Cañas, María Paula Toro-Gómez, Larissa Sayuri Moreira Sugai, Hernán Darío Benítez Restrepo, Jorge Rudas, Breyner Posso Bautista, Luís Felipe Toledo, Simone Dena, Adão Henrique Rosa Domingos, Franco Leandro de Souza, Selvino Neckel-Oliveira, Anderson da Rosa, Vítor Carvalho-Rocha, José Vinícius Bernardy, José Luiz Massao Moreira Sugai, Carolina Emília dos Santos, Rogério Pereira Bastos, Diego Llusia, and Juan Sebastián Ulloa. A dataset for benchmarking Neotropical anuran calls identification in passive acoustic monitoring. *Scientific Data*, 10(1):1–12, 2023. ISSN 20524463. doi: 10.1038/s41597-023-02666-2.

[16] Anurag Kumar and Bhiksha Raj. Deep cnn framework for audio event recognition using weakly labeled web data, 2022. URL `https://arxiv.org/abs/1707.02530`.

[17] Peter Foster, Siddharth Sigtia, Sacha Krstulovic, Jon Barker, and Mark D. Plumbley. Chime-home: A dataset for sound source recognition in a domestic environment. *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA 2015*, pages 1–5, 2015. doi: 10.1109/WASPAA.2015.7336899.

[18] Jort F. Gemmeke, Daniel P.W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio Set: An ontology and human-labeled dataset for audio events. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 776–780, 2017. ISSN 15206149. doi: 10.1109/ICASSP.2017.7952261.

[19] Ankit Shah, Anurag Kumar, Alexander G. Hauptmann, and Bhiksha Raj. A closer look at weak label learning for audio events, 2018. URL `https://arxiv.org/abs/1804.09288`.

[20] Shuyang Zhao, Toni Heittola, and Tuomas Virtanen. Active Learning for Sound Event Detection. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 28:2895–2905, 2020. ISSN 23299304. doi: 10.1109/TASLP.2020.3029652.

[21] Yooju Shin, Susik Yoon, Sundong Kim, Hwanjun Song, Jae Gil Lee, and Byung Suk Lee. Coherence-Based Label Propagation Over Time Series for Accelerated Active Learning. *ICLR 2022 - 10th International Conference on Learning Representations*, 2022.

[22] Heinrich Dinkel, Zhiyong Yan, Yongqing Wang, Junbo Zhang, and Yujun Wang. Pseudo Strong Labels for Large Scale Weakly Supervised Audio Tagging. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2022-May:336–340, 2022. ISSN 15206149. doi: 10.1109/ICASSP43922.2022.9746431.

[23] Nicolas Turpault, Romain Serizel, Emmanuel Vincent, Nicolas Turpault, Romain Serizel, and Emmanuel Vincent. Analysis of weak labels for sound event tagging. 2021. URL `https://hal.inria.fr/hal-03203692`.

[24] Ivo Trowitzsch, Jalil Taghia, Youssef Kashef, and Klaus Obermayer. The NIGENS General Sound Events Database. Technical report, Technische Universität Berlin, 2020. arXiv:1902.08314 [cs.SD].

[25] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. Metrics for polyphonic sound event detection. *Applied Sciences (Switzerland)*, 6(6), 2016. ISSN 20763417. doi: 10.3390/app6060162.

1 Appendix

1.1 Proof of theorem 1

Proof. From figure 2 we note that we can divide the integral in Eq 4.4 into five different regions:

1. $F(e_t, q, \gamma) = 0$ when $t \in [t_0, t_1]$,
2. $F(e_t, q, \gamma) = f(e_t, q)$ when $t \in [t_1, t_2]$ (light blue),

3. $F(e_t, q, \gamma) = f_{\max}$ when $t \in [t_2, t_3]$ (dark blue),
4. $F(e_t, q, \gamma) = f(e_t, q)$ when $t \in [t_3, t_4]$ (light blue), and
5. $F(e_t, q, \gamma) = 0$ when $t \in [t_4, t_5]$.

The area under the curve for (1) and (5) is 0, and for (2) and (4) it is the light blue area in figure 2 which are by symmetry equal, and the area under (3) is the dark blue area in figure 2, which means that Eq. 4.4 can be written as

$$P(e, q, \gamma) = \frac{1}{d_e + d_q} \left[2 \int_{t_1}^{t_2} f(e_t, q) \mathrm{d}t + \int_{t_2}^{t_3} f_{\max} \mathrm{d}t \right]. \tag{1.1}$$

The QIoU $f(e_t, q)$ is defined in Eq 4.2, and now we only need to define $t_0, \ldots, t_3, f_{\min}, f_{\max}$ as functions of e, q and γ to compute Eq 1.1. We will do this with the help of figure 2 and figure 13.

In figure 13 we can see the states $t_0, \ldots, t_3$ illustrated. The initial state $t = t_0$ is when the event e_t aligns with the query q which happens at $t_0 = 0$.

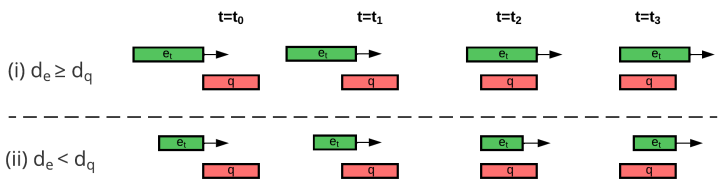


Figure 13: An illustration of how the sound event e_t and the query segment q overlap at the four distinct states at times $t = t_0, \ldots, t_3$ (see figure 2). (For brevity we do not show state t_4 which is symmetrical to t_1 and t_5 which is symmetrical to t_0.)

State $t = t_1$. The shift we see at t_1 in figure 2 is when $F(e, q, \gamma)$ goes from being 0 to $f(e_t, q)$, i.e., the time t where there is enough overlap between the sound event and the query segment for the annotator criterion to be minimally fulfilled. This happens when $h(e_t, q) = \gamma$ which by Eq. 3.1 means that $|e_{t_1} \cap q| = \gamma d_e$ and therefore we know that $t_1 - t_0 = \gamma d_e$ (see $t = t_1$ in figure 13).

State $t = t_2$. The shift we see at t_2 in figure 2 is where the maximum QIoU $f_{\max}$ is achieved for the first time. The time t_2 when this happens and the value $f_{\max}$ depends on whether the event length is long enough to cover the entire query or not. We therefore need to consider case (i) when $d_e \geq d_q$ and case (ii) when $d_e < d_q$. If $d_e \geq d_q$ then $f_{\max}$ is achieved when the end of the sound event aligns with the end of the query which happens when the

sound event has moved the whole length of the query at $t_2^{(i)} = d_q$ (see $t = t_2$ case (i) in figure 13) and thus $t_2^{(i)} - t_1^{(i)} = d_q - \gamma d_e = \alpha^{(i)}$. If $d_e < d_q$ then $f_{\max}$ is achieved when the beginning of the sound event aligns with the beginning of the query (see $t = t_2$ case (ii) in figure 13) which happens when the sound event has moved the whole length of the sound event $t_2^{(ii)} = d_e$ and thus $t_2^{(ii)} - t_1^{(ii)} = d_e - \gamma d_e = \alpha^{(ii)}$.

State $t = t_3$. This shift we see at t_3 in figure 2 is when $f_{\max}$ is achieved for the last time. Again we need to consider the two cases, and in case (i) this happens after the event has moved the whole event length at $t_3^{(i)} = d_e$, and in case (ii) this happens when the event has moved the whole query length at $t_3^{(ii)} = d_q$. We can therefore write $t_3^{(i)} - t_2^{(i)} = d_e - d_q = \beta^{(i)}$ for case (i), and $t_3^{(ii)} - t_2^{(ii)} = d_q - d_e = \beta^{(ii)}$ for case (ii).

For convenience we introduce the helper variables

$$\alpha = \begin{cases} \alpha^{(i)}, \text{ if } d_e \geq d_q, \text{ and} \\ \alpha^{(ii)}, \text{ if } d_e < d_q, \end{cases} \tag{1.2}$$

$$\beta = \begin{cases} \beta^{(i)}, \text{ if } d_e \geq d_q, \text{ and} \\ \beta^{(ii)}, \text{ if } d_e < d_q, \end{cases} \tag{1.3}$$

to hide the dependence on the cases for now and define $t_0 = 0$, $t_1 = d_e\gamma$, $t_2 = t_1 + \alpha$, and $t_3 = t_2 + \beta$. Finally we note that

$$f_{\max} = \begin{cases} 1, \text{ if } d_e \geq d_q, \\ \frac{d_e}{d_q}, \text{ if } d_e < d_q, \end{cases} \tag{1.4}$$

since $F(e_t, q, \gamma) = 1$ for $t \in [t_2, t_3]$ in case (i), and otherwise $F(e_t, q, \gamma) = \frac{d_e}{d_q}$ since $|q \cap e_t| = d_e$ between t_3 and t_4, and that

$$f_{\min} = \frac{\gamma d_e}{d_q}, \tag{1.5}$$

which is always the QIoU when Eq 3.2 is minimally fulfilled.

We can now compute each part of Eq 1.1 separately. We start by observing that $f(e_t, q)$ is a linear function between $t = t_1$ and $t = t_2$ going from $f_{\min}$ to $f_{\max}$, which means that we can compute

$$\int_{t_1}^{t_2} f(e_t, q) \mathrm{d}t = \int_0^{\alpha} f(e_{t+t_1}, q) \mathrm{d}t, \tag{1.6}$$

$$= \frac{\alpha(f_{\min} + f_{\max})}{2}, \tag{1.7}$$

since $t_2 - t_1 = \alpha$ by definition. The other integral is over $f_{\max}$ between t_2 and t_3 where $t_3 - t_2 = \beta$ by definition which give

$$\int_{t_2}^{t_3} f_{\max} \mathrm{d}t = \int_0^{\beta} f_{\max} \mathrm{d}t \tag{1.8}$$

$$= \beta f_{\max} \tag{1.9}$$

Finally we get

$$P(e, q, \gamma) = \frac{1}{d_e + d_q} \left[\alpha(f_{\min} + f_{\max}) + \beta f_{\max} \right], \tag{1.10}$$

which is the expected QIoU of queries of event length d_q for sound events of length d_e when the annotator criterion is γ. With a slight abuse of notation we will use $P(e, q, \gamma) = P(d_e, d_q, \gamma)$ since the resulting expression in Eq 1.10 is only a function of d_e, d_q and γ.

Next we consider case (i) and case (i) separately by substituting the convenience variables α, β, $f_{\min}$, and $f_{\max}$ for the actual values in each case (i) and (ii).

Case (i). $d_e \geq d_q$.

$$P(d_e, d_q, \gamma) = \frac{1}{d_e + d_q} \left[\frac{2\alpha\gamma d_e + \alpha^2}{d_q} + \beta f_{\max} \right] \tag{1.11}$$

$$= \frac{1}{d_e + d_q} \left[\frac{2(d_q - \gamma d_e)\gamma d_e + (d_q - \gamma d_e)^2}{d_q} + (d_e - d_q) \right] \tag{1.12}$$

$$= \frac{1}{d_e + d_q} \left[d_e - \frac{d_e^2 \gamma^2}{d_q} \right] \tag{1.13}$$

Case (ii). $d_e < d_q$.

$$P(d_e, d_q, \gamma) = \frac{1}{d_e + d_q} \left[\frac{2\alpha\gamma d_e + \alpha^2}{d_q} + \beta f_{\max} \right] \tag{1.14}$$

$$= \frac{1}{d_e + d_q} \left[\frac{2(d_e - \gamma d_e)\gamma d_e + (d_e - \gamma d_e)^2}{d_q} + (d_q - d_e)\frac{d_e}{d_q} \right] \tag{1.15}$$

$$= \frac{1}{d_e + d_q} \left[d_e - \frac{d_e^2 \gamma^2}{d_q} \right] \tag{1.16}$$

Since both cases yield the same expression we can simply describe the expected QIoU as

$$P(d_e, d_q, \gamma) = \frac{1}{d_e + d_q} \left[d_e - \frac{d_e^2 \gamma^2}{d_q} \right] \tag{1.17}$$

which thus describes the expected QIoU of an annotator with annotator criterion γ for an arbitrary query q and a sound event e_t placed uniformly at random at time $t \in [0, T - d_e]$.

□

Paper B

From weak to strong sound event labels using adaptive change-point detection and active learning

John Martinsson[1,2], Olof Mogren[1], Maria Sandsten[2], Tuomas Virtanen[3]

[1] *Computer Science, RISE Research Institutes of Sweden*

[2] *Centre for Mathematical Sciences, Lund University*

[3] *Signal Processing Research Centre, Tampere University*

Abstract

We propose an adaptive change point detection method (A-CPD) for machine guided weak label annotation of audio recording segments. The goal is to maximize the amount of information gained about the temporal activations of the target sounds. For each unlabeled audio recording, we use a prediction model to derive a probability curve used to guide annotation. The prediction model is initially pre-trained on available annotated sound event data with classes that are disjoint from the classes in the unlabeled dataset. The prediction model then gradually adapts to the annotations provided by the annotator in an active learning loop. We derive query segments to guide the weak label annotator towards strong labels, using change point detection on these probabilities. We show that it is possible to derive strong labels of high quality with a limited annotation budget, and show favorable results for A-CPD when compared to two baseline query segment strategies.

Proceedings: In proceedings of the 32nd European Signal Processing Conference (EUSIPCO), Lyon, France, 2024. (Nominated for best student paper.)

Keywords: Active learning, annotation, sound event detection, deep learning

1 Introduction

Most audio datasets today consists of weakly labeled data with imprecise timing information [1], and there is a need for efficient and reliable annotation processes to acquire labels with precise timing information. We refer to such labels as strong labels. The performance of sound event detection (SED) models improve with strong labels [2], and strong labels become especially important when we want to count the number of occurrences of an event class. For example in bioacoustics, where counting the number of vocalizations of an animal species can be used to estimate population density and draw ecological insights [3].

Crowdsourcing the strong labels is challenging and an attractive solution is to crowdsource weak labels to enable reconstruction of the strong labels [4, 5]. Asking the annotator for strong labels requires more work and it can in the worst case lead to the annotator misunderstanding the task [5].

Disagreement-based active learning is the most used form of active learning for sound event detection [6, 7, 8, 9], focusing on selecting what audio segment to label next. The recordings are either split into equal length audio segments [6, 7, 9] or segments depending on the structure of the sound [8]. Each segment is then given a weak label by the annotator.

We use a weak label annotator to derive strong labels as in [5], but instead of using fixed length query segments we adapt the query segments to the data, in the setting of active learning. We propose an adaptive change point detection (A-CPD) method which splits a given audio recording into a set of audio segments, or queries. The queries are then labeled by the annotator and the strong labels are derived and evaluated.

See Fig. 1 for an illustration where a set of seven queries are used either optimally or suboptimally for a given audio recording with three sound events. We assume three sound events to be detected in each audio recording as a simplification during method development. We aim to adapt the set of queries in such a way that the information about the temporal activations of the target sounds is maximized. Note that we aim to actively guide the annotator during the annotation of the audio recordings, rather than actively choose which audio recordings to annotate which is typically done in active learning.

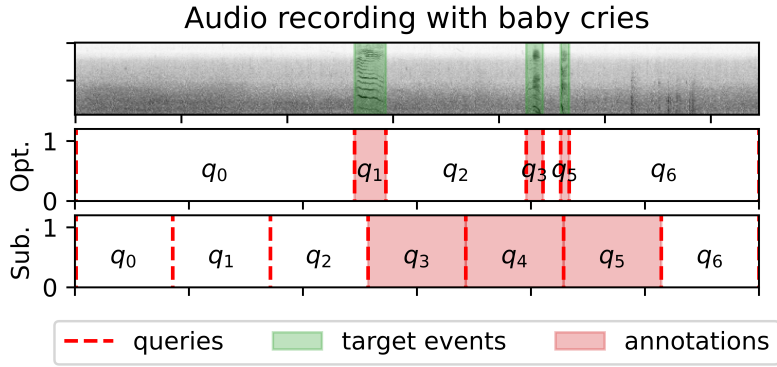


Figure 1: Illustration of segmentation of an audio spectrogram with three target events shown in shaded green (top panel) into a set of audio query segments $q_0, \ldots, q_6$ using an optimal method w.r.t the derived strong label timings (middle panel) and a sub-optimal method (bottom panel). Resulting annotations, from the weak labels given by the annotator, are shown in shaded red for both methods. Query q_4 for the optimal method is omitted for clarity.

2 Sound event annotation using active learning

We consider SED tasks where the goal is to predict the presence of a given target event class. The results can also be generalized into the multi-class setting. Given a restricted annotation budget and no initial labels we aim to derive strong labels using active learning to train a SED system. To this end, we propose the following machine guided annotation process.

Let $\mathcal{D}_L^{(k)}$ denote the set of labeled audio recordings and $\mathcal{D}_U^{(k)}$ the set of unlabeled audio recordings at active learning iteration k. Further, let $\mathcal{A}^{(k)} = \{(s_i^{(j)}, e_i^{(j)}, c_i^{(j)})\}_{i=1}^{B}{}_{j=1}^{k}$ denote the annotations of segments, where s denotes the onset, e the offset, and $c \in \{0, 1\}$, the weak label for each segment i of the B annotated segments in audio recording j.

We start without any labels, $\mathcal{A}^{(0)} = \mathcal{D}_L^{(0)} =$, and all audio recordings are unlabeled, $\mathcal{D}_U^{(0)} = \{\mathbf{x}_j\}_{j=1}^{N}$, where $\mathbf{x}_j \in \mathbb{R}^T$ denotes an audio recording of length T, and N denotes the total number of audio recordings. We then loop for each $k \in \{1, \ldots, N\}$ and:

1. choose a random unlabeled audio recording $\mathbf{x}$ from $\mathcal{D}_U^{(k-1)}$,
2. derive a set of B audio query segments $Q = \{q_i\}_{i=0}^{B-1}$ using a query strategy where $q_i = (s_i, e_i)$ consists of the start s_i and end e_i timings for query i,
3. send the queries to the annotator (returning a weak label for each query) and add the annotations to the set of segment labels $\mathcal{A}^{(k)} = \mathcal{A}^{(k-1)} \cup \{(s_i, e_i, c_i)\}_{i=1}^{B}$,

4. *In case of A-CPD*: use the annotations $\{(s_i, e_i, c_i)\}_{i=1}^{B}$ to update the query strategy, and

5. update the labeled recording set $\mathcal{D}_L^{(k)}$ by adding $\mathbf{x}$ and the unlabeled recording set $\mathcal{D}_U^{(k)}$ by removing $\mathbf{x}$.

For brevity we have omitted the dependence on k for $\mathbf{x}_{r_k}$ and $(s_i^{(r_k)}, e_i^{(r_k)}, c_i^{(r_k)})$ in the description of the annotation loop, where $r_k \in \{1, \ldots, N\}$ would denote the randomly sampled audio recording for iteration k. After the annotation loop all N audio recordings have been annotated exactly once with the query method used in step (2), resulting in a set of annotations $\mathcal{A}^{(N)} = \{(s_i^{(j)}, e_i^{(j)}, c_i^{(j)})\}_{i=1\,j=1}^{B\,N}$.

Note that B is not the number of sound events in the recording, but the number of query segments allowed when annotating the recording. The smallest number of query segments to derive the ground truth strong labels does, however, depend on the number of sound events M in the recording as $2M + 1$ (see Section 3.4). A-CPD is developed to provide strong labels using as few as $B = 2M + 1$ queries.

The total annotation budget used will scale with both N and B. Typically we would aim to reduce N by actively sampling the data points to annotate, but we instead aim to reduce B. Think of B as a part of the annotation cost of an audio recording, which can be reduced with maintained label strength by guiding the annotator during the annotation process.

3 Query strategies

In this section we describe the studied query strategies.

3.1 The adaptive change point detection strategy (A-CPD)

To produce a set of queries for a given audio recording $\mathbf{x}$ at annotation round k we perform three key steps:

1. update a prediction model using the annotations from round $k - 1$ (initialized with pre-training if $k = 0$),

2. predict probabilities indicating the presence of the target class in the recording using the model, and

3. apply change point detection to the probabilities to derive the queries.

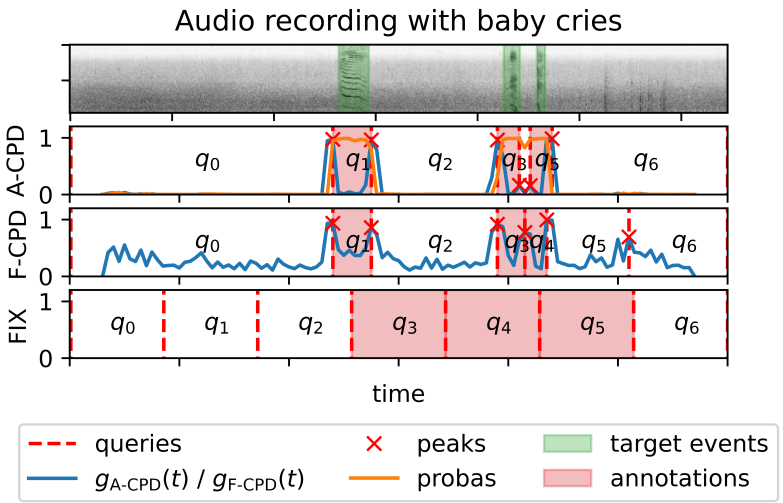


Figure 2: Qualitative example of how the different query strategies A-CPD, F-CPD and FIX segment a spectrogram of an audio recording with three target events shown in shaded green (top panel) into $B = 7$ queries. A-CPD (second panel) uses change point detection (blue line) on the probability curve from a prediction model (orange line) to detect the $B - 1$ most prominent peaks (red crosses) which are used to construct a set of queries $\{q_0, \ldots, q_{B-1}\}$ (dashed red lines). Each query $q_i = (s_i, e_i)$ is given a weak label $c_i \in \{0, 1\}$ ($c = 1$ shown as shaded red), resulting in the i:th annotation (s_i, e_i, c_i). F-CPD (third panel) uses change point detection directly on the cosine distances in embedding space (blue line) and thereafter constructs queries in the same way as A-CPD. FIX (fourth panel) uses fixed length queries.

The pre-training of the prediction model can be done in a supervised or unsupervised way. The important property is that the model reacts to changes in the audio recording related to the presence or absence of the target class. However, it is not strictly necessary that the model reacts *only* to those changes.

Let $h_k : \mathbb{R}^L \to [0, 1]$ denote a model that predicts the probability of an audio segment of length L belonging to the target event class. In principle, any prediction model can be used. For a given audio recording $\mathbf{x}$ the prediction model $h_k(\cdot)$ is applied to consecutive audio segments to derive a probability curve, shown as the orange curve for A-CPD in Fig. 2. The consecutive audio segments are derived using a moving window of L seconds with hop size $L/4$.

We define the Euclidean distance between two points $t - \alpha$ and $t + \alpha$ on the probability curve as:

$$g^{(k)}_{\text{A-CPD}}(t) = ||h_k(t - \alpha) - h_k(t + \alpha)||, \tag{3.1}$$

shown as the blue curve for A-CPD in Fig. 2. The previous probability is compared with the next probability in Eq. 3.1, and $\alpha = L/4$ (hop size) is therefore chosen to ensure a 50% overlap between the audio segments for these probabilities.

Let t be a local optimum of $g^{(k)}_{\text{A-CPD}}(t)$, and all such local optima are called peaks. We rank peaks based on *prominence*. For any given peak t, let t_l and t_r denote the closest local

minima of $g_k(\cdot)$ to the left and right of t. The prominence of the peak at t is defined as $|g_k(t) - \max(g_k(t_l), g_k(t_r))|$. Let $\mathcal{T}_{\text{A-CPD}} = \{t_1, t_2, \ldots, t_{B-1}\}$ be the $B-1$ most prominent peaks of a given audio recording such that $t_1 \leq t_2 \leq \cdots \leq t_{B-1}$, shown as red crosses in Fig. 2. The A-CPD query method is then defined as:

$$Q^{(k)}_{\text{A-CPD}} = \{(0, t_1), (t_1, t_2), \ldots, (t_{B-1}, T)\}, \tag{3.2}$$

which are shown as dashed red lines in Fig. 2, where T is the length of the audio recording and B is the number of queries used. Note that $g^{(k)}_{\text{A-CPD}}(t)$ will gradually become more sensitive towards changes between presence and absence of the target class in the recording with additional annotations, and become less sensitive to other unrelated changes.

3.2 The fixed change point detection strategy (F-CPD)

The fixed change point detection (F-CPD) method used as a reference derives the queries by computing the cosine distance between the previous embedding at time $t - \alpha$ and the next embedding at time $t + \alpha$:

$$g_{\text{F-CPD}}(t) = 1 - \frac{\mathbf{e}_{t-\alpha} \cdot \mathbf{e}_{t+\alpha}}{||\mathbf{e}_{t-\alpha}||||\mathbf{e}_{t+\alpha}||}, \tag{3.3}$$

where $\mathbf{e}_t = f_\theta(\mathbf{x}_t)$ denotes the embedding of consecutive audio segments $\mathbf{x}_t$ centered at second t using the embedding function $f_\theta : \mathbb{R}^L \to \mathbb{R}^K$. The cosine distance curve for an audio recording is shown as the blue line for F-CPD in Fig. 2. This method is similar to [8] except that embeddings are derived for 1.0 seconds of audio instead of 0.02. We therefore directly compare the previous and next embeddings instead of a moving average as in [8].

The most prominent peaks in the cosine distance curve is then selected, $\mathcal{T}_{\text{FIX}} = \{t_1, t_2, \ldots, t_{B-1}\}$, and the set of queries are defined as in Eq. 3.2, shown as dashed red lines for F-CPD in Fig 2.

3.3 The fixed length strategy (FIX)

In the fixed length query strategy (FIX) audio is split into equal length segments and then labeled. Let $d = T/B$, then the queries are defined as

$$Q_{\text{FIX}} = \{(0d, 1d), (1d, 2d), \ldots, ((B-1)d, Bd)\}, \tag{3.4}$$

shown as dashed red lines for FIX in Fig 2. This is the setting most previous active learning work for SED consider.

3.4 The oracle strategy (ORC)

The oracle query strategy constructs the queries based on the ground truth presence and absence annotations

$$Q_{\text{ORC}} = \{(s_0, e_0), (s_1, e_1), \ldots, (s_{B_{\text{suff}}-1}, e_{B_{\text{suff}}-1})\}, \tag{3.5}$$

where (s_i, e_i) is the onset and offset for segment i where the target event is either present or not. B_{suff} is the sufficient number of queries to get the true strong labels, which relate to the number of target events M in the given audio recording by $B_{\text{suff}} = 2M + 1$. ORC is undefined for $B < B_{\text{suff}}$.

3.5 The role of query strategies in the annotation process

The query strategies described in this section are then used in step (2) of the annotation loop described in Section 2. Note that when the queries are not adapted to the audio recording multiple events can end up being counted as one. In Fig. 2 we can see this for F-CPD where q_3 and q_4 are directly adjacent, meaning that they are not resolved as two separate events, and for FIX where q_3, q_4 and q_5 are all directly adjacent. A-CPD often resolves all three events. Fig 2 is a qualitative example of all three methods, and quantitative results to further support this claim are provided later in table 1.

The FIX length query segments depend on the query timings and target event timings aligning by chance since the query construction is independent of the target events. The A-CPD method aim to create query segments that are aligned with the target events by construction. In addition, the number of queries needed to derive the strong labels scale with the number of target events in the recording for A-CPD, which can be beneficial.

4 Evaluation

4.1 Datasets

We create three SED datasets for evaluation, each with a different target event class: Meerkat, Dog or Baby cry. The Meerkat sounds are from the DCASE 2023 few-shot bioacoustic SED dataset [10] and the Dog and Baby cry sounds from the NIGENS dataset [11]. The sounds used for absence of an event are from the 15 background types in the TUT Rare sound events dataset [12].

The audio recordings in each dataset are created by randomly selecting $M = 3$ sound events from that event class and mixing them together with a randomly selected background

recording of length $T = 30$ seconds. In this way we know that exactly $B_{\text{suff}} = 2M+1 = 7$ queries are *sufficient* and *necessary* to derive the ground truth strong labels using a weak label annotator. The mixing is done using Scaper [13] at an SNR of 0 dB. In total we generate $N = 300$ audio recordings using this procedure for each event class as training data and equally many as test data.

The source files used in the mixing uses the supplied splits in [11] and [12], except for the Meerkat sounds where non exist and the split is done on a recording level.

4.2 Evaluation metrics

We evaluate the methods by annotating the mixed training datasets using the query strategies described in Section 2 and the annotation loop described in Section 3. The quality of the annotations are then measured in two ways: (i) how strong the annotations are compared to the ground truth, and (ii) the test time performance of two evaluation models trained using the different annotations.

The evaluation metrics used in case (i) and (ii) are event-based F_1-score (F_{1e}) and segment-based F_1-score (F_{1s}) [14]. The segment size for F_{1s} is set to 0.05 seconds, and the collar for F_{1e} is set to 0.5 seconds. In case (i) the F_{1s} measures how much of the audio that has been correctly labeled and in case (ii) F_{1s} measures how much of the audio that has been correctly predicted by the evaluation model. The F_{1e} score is only used to measure how close the annotations are to the ground truth labels in the training data.

Annotator model Let $\mathcal{A}_{gt}^{(j)} = \{(s_i, e_i, c = 1)\}_{i=1}^{3}$ denote the set of ground truth target event labels for audio recording j, where s_i is the onset, e_i the offset and $c = 1$ indicate the presence of the target event.

We use $\mathcal{A}_{gt}^{(j)}$ to simulate an annotator for recording j. For a given query segment we check the overlap ratio with the ground truth target event labels. Formally, if there exists an annotation $(s_i, e_i, c_i = 1)$ s.t.

$$\frac{(s_i, e_i) \cap (s_q, e_q)}{|s_i - e_i|} \geq \gamma, \tag{4.1}$$

holds for the given query segment $q = (s_q, e_q)$, then the annotator returns $c_i = 1$ for query q, and $c_i = 0$ otherwise. Annotation noise is added by flipping the returned label with probability β. In this work $\gamma = 0.5$, and $\beta \in \{0.0, 0.2\}$.

4.3 Implementation details and experiment setup

Prediction model The prediction model $h_k(\cdot)$ is modeled using a prototypical neural network (ProtoNet) [15]. The prototypes are easily updated at each annotation round k using a running average between each previous prototype and the newly labeled audio embeddings. We model the embedding function $f_\theta(\cdot)$ using BirdNET [16], a convolutional neural network pre-trained on large amounts of bird sounds.

Evaluation models We use two models to evaluate the test time performance of models trained on the annotations obtained using each query strategy: a two layer multilayer perceptron (MLP) and a ProtoNet. The MLP is trained using the Adam optimizer and cross-entropy loss. Each query strategy is run 10 times and the evaluation models are trained on the embeddings using the resulting labeled datasets. ProtoNet is used in two ways: as a prediction model in the proposed A-CPD method, and as an evaluation model.

4.4 Results

In Table 1 we show the average F_{1s}-score and F_{1e}-score for the training data annotations over 10 runs for each dataset and with the sufficient nu $B = B_{\text{suff}} = 7$. The A-CPD method outperforms the other methods for all studied target event classes. The standard deviation is in all cases less than 0.03 (omitted from table for brevity), and the baseline query strategies are deterministic when $\beta = 0$.

Table 1: Average F_{1s}-score and F_{1e}-score for the training annotations for each annotation process and target event class with $\beta = 0$

Strategy	Meerkat		Dog		Baby	
	F_{1s}	F_{1e}	F_{1s}	F_{1e}	F_{1s}	F_{1e}
ORC	1.00	1.00	1.00	1.00	1.00	1.00
A-CPD	0.31	0.57	0.29	0.45	0.62	0.60
F-CPD	0.16	0.44	0.21	0.30	0.48	0.45
FIX	0.11	0.00	0.19	0.00	0.41	0.01

In Fig. 3 we show the average F_{1s}-score over all runs and event classes for the annotations derived from each query strategy. The proposed A-CPD method has a strictly higher F_{1s}-score than the FIX and F-CPD baselines for all budgets and noise settings. We also see that there is still a significant gap to the ORC strategy. The noisy annotator ($\beta = 0.2$) drastically reduce the label quality for all studied strategies, especially ORC dropping from an F_{1s}-score of 1.0 (omitted from figure) to ≈ 0.28 (large drop due to class-imbalance).

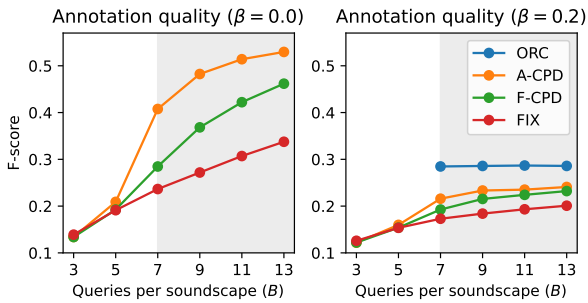


Figure 3: The average F_{1s}-score over the three classes for each of the studied annotation processes plotted against the number of queries per audio recording, B. The results are shown for an annotator without noise (left) and with $\beta = 0.2$ (right). Note that ORC is 1.0 when $\beta = 0$ and is therefore not shown in the left figure. Shaded region where $B \geq B_{\text{suff}}$.

In Fig. 4 we show the average test time F_{1s}-score of a ProtoNet (top) and a MLP (bottom) trained using the annotations from each of the studied annotation strategies and settings. The A-CPD method outperforms the other methods when $B \geq 7$. For the ProtoNet the FIX method outperform A-CPD when $B < 7$ and for the MLP the results are similar.

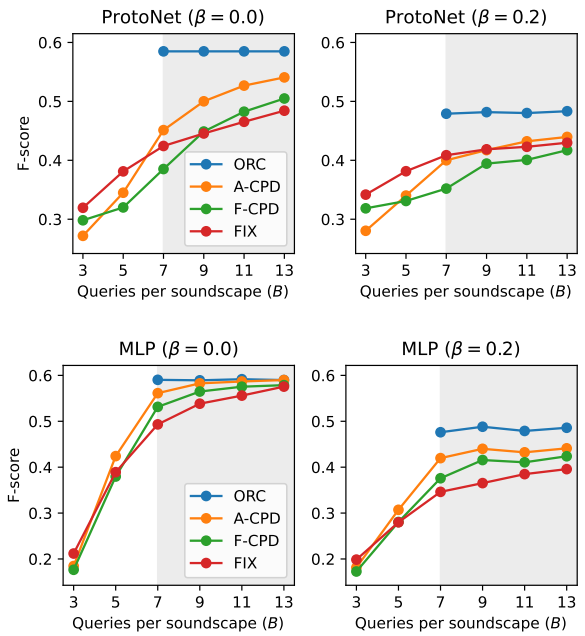


Figure 4: The average test time F_{1s}-score over the studied sound classes for a ProtoNet (top) and the MLP (bottom) trained with the annotations from each respective annotation process and setting. Shaded region where $B \geq B_{\text{suff}}$.

Table 2 and 3 show the average F_{1s}-score and standard deviation for the three different

event classes for all studied query strategies. The average is over 10 runs, and the number of queries is set to $B = 7$. Table 2 shows the F_{1s}-score for the ProtoNet evaluation model. A-CPD achieves a higher F_{1s}-score for the meerkat and baby datasets. On average A-CPD outperforms the other methods as seen in Fig. 4. Table 3 shows the F_{1s}-score for the MLP evaluation model. A-CPD achieves a higher F_{1s}-score for all studied datasets.

Table 2: Average test time F_{1s}-score for ProtoNet with $\beta = 0$.

Strategy	Meerkat	Dog	Baby
ORC	0.46	0.48	0.81
A-CPD	0.44 ± 0.00	0.20 ± 0.01	0.71 ± 0.02
F-CPD	0.31	0.19	0.66
FIX	0.34	0.25	0.68

Table 3: Average test time F_{1s}-score for MLP with $\beta = 0$.

Strategy	Meerkat	Dog	Baby
ORC	0.43 ± 0.00	0.51 ± 0.01	0.83 ± 0.00
A-CPD	0.44 ± 0.00	0.43 ± 0.02	0.81 ± 0.01
F-CPD	0.38 ± 0.01	0.42 ± 0.02	0.79 ± 0.01
FIX	0.33 ± 0.02	0.40 ± 0.02	0.75 ± 0.02

4.5 Discussion

The results in all tables are for the sufficient budget $B = B_{\text{suff}} = 2M+1$. In practice we do not know B_{suff}. However, the A-CPD method is applicable also for an arbitrary number of sound events in the recording when B is chosen sufficiently large. This choice need to be made for all the studied methods. We show the benefit of A-CPD for differently chosen B in Fig. 3. Estimating B_{suff} based on the audio recording could further reduce the number of queries used and is left as future work.

We chose $\gamma = 0.5$ in the annotator model since the annotator should be able to detect a target event if more than 50% of the event occurs within the query segment. This choice is however non-trivial, and depends on the expertise of the annotator and target class among others. We observe similar results on average as those presented in the paper for $\gamma \in \{0.05, 0.25, 0.5, 0.75, 0.95\}$ (not shown).

We use BirdNET [16] to model the embedding function since we study bioacoustic target classes. However, an embedding function such as PANNs [17] may also be used if the target classes are more general.

5 Conclusions

We have presented a query strategy based on adaptive change point detection (A-CPD) which derive strong labels of high quality from a weak label annotator in an active learning setting. We show that A-CPD gives strictly stronger labels than all other studied baseline query strategies for all studied budget constraints and annotator noise settings. We also show that models trained using annotations from A-CPD tend to outperform models trained with the weaker labels from the baselines at test time. We note that the gap to the oracle method is still large, leaving room for improvements in future work.

References

[1] Qiuqiang Kong, Yong Xu, Wenwu Wang, and Mark D. Plumbley. Sound event detection of weakly labelled data with cnn-transformer and automatic threshold optimization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28: 2450–2460, 2020. doi: 10.1109/TASLP.2020.3014737.

[2] Shawn Hershey, Daniel P.W. Ellis, Eduardo Fonseca, Aren Jansen, Caroline Liu, R. Channing Moore, and Manoj Plakal. The benefit of temporally-strong labels in audio event classification. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 366–370, 2021. ISSN 15206149. doi: 10.1109/ICASSP39728.2021.9414579.

[3] Tiago A. Marques, Len Thomas, Stephen W. Martin, David K. Mellinger, Jessica A. Ward, David J. Moretti, Danielle Harris, and Peter L. Tyack. Estimating animal population density using passive acoustics. *Biological Reviews*, 88(2):287–309, 2013. doi: https://doi.org/10.1111/brv.12001. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/brv.12001`.

[4] Irene Martin-Morato, Manu Harju, and Annamaria Mesaros. Crowdsourcing Strong Labels for Sound Event Detection. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 246–250, 2021. ISSN 19471629. doi: 10.1109/WASPAA52581.2021.9632761.

[5] Irene Martin-Morato and Annamaria Mesaros. Strong Labeling of Sound Events Using Crowdsourced Weak Labels and Annotator Competence Estimation. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 31:902–914, 2023. ISSN 23299304. doi: 10.1109/TASLP.2022.3233468.

[6] Zhao Shuyang, Toni Heittola, and Tuomas Virtanen. Active learning for sound event classification by clustering unlabeled data. *ICASSP, IEEE International Conference*

on Acoustics, Speech and Signal Processing - Proceedings, pages 751–755, 2017. ISSN 15206149. doi: 10.1109/ICASSP.2017.7952256.

[7] Zhao Shuyang, Toni Heittola, and Tuomas Virtanen. An active learning method using clustering and committee-based sample selection for sound event classification. *16th International Workshop on Acoustic Signal Enhancement, IWAENC 2018 - Proceedings*, pages 116–120, 2018. doi: 10.1109/IWAENC.2018.8521336.

[8] Zhao Shuyang, Toni Heittola, and Tuomas Virtanen. Active Learning for Sound Event Detection. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 28:2895–2905, 2020. ISSN 23299304. doi: 10.1109/TASLP.2020.3029652.

[9] Yu Wang, Mark Cartwright, and Juan Pablo Bello. Active Few-Shot Learning for Sound Event Detection. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 1551–1555, 2022. ISSN 19909772. doi: 10.21437/Interspeech.2022-10907.

[10] I. Nolasco, S. Singh, E. Vidana-Villa, E. Grout, J. Morford, M. Emmerson, F. Jensens, H. Whitehead, I. Kiskin, A. Strandburg-Peshkin, L. Gill, H. Pamula, V. Lostanlen, V. Morfi, and D. Stowell. Few-shot bioacoustic event detection at the DCASE 2022 challenge. (November):1–5, 2022.

[11] Ivo Trowitzsch, Jalil Taghia, Youssef Kashef, and Klaus Obermayer. The NIGENS General Sound Events Database. Technical report, Technische Universität Berlin, 2020. arXiv:1902.08314 [cs.SD].

[12] Aleksandr Diment, Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. TUT Rare sound events, Development dataset, January 2018. URL `https://doi.org/10.5281/zenodo.401395`.

[13] Justin Salamon, Duncan MacConnell, Mark Cartwright, Peter Li, and Juan Pablo Bello. Scaper: A library for soundscape synthesis and augmentation. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 344–348, 2017. doi: 10.1109/WASPAA.2017.8170052.

[14] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. Metrics for polyphonic sound event detection. *Applied Sciences (Switzerland)*, 6(6), 2016. ISSN 20763417. doi: 10.3390/app6060162.

[15] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, pages 4078–4088, 2017. ISSN 10495258.

[16] Stefan Kahl, Connor M. Wood, Maximilian Eibl, and Holger Klinck. BirdNET: A deep learning solution for avian diversity monitoring. *Ecological Informatics*, 61 (January):101236, 2021. ISSN 15749541. doi: 10.1016/j.ecoinf.2021.101236.

[17] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28: 2880–2894, 2020. doi: 10.1109/TASLP.2020.3030497.

Paper C

DMEL: the differentiable log-Mel spectrogram as a trainable layer in neural networks

John Martinsson[1,2], Maria Sandsten[2]

[1] *Computer Science, RISE Research Institutes of Sweden*
[2] *Centre for Mathematical Sciences, Lund University*

Abstract

In this paper we present the differentiable log-Mel spectrogram (DMEL) for audio classification. DMEL uses a Gaussian window, with a window length that can be jointly optimized with the neural network. DMEL is used as the input layer in different neural networks and evaluated on standard audio datasets. We show that DMEL achieves a higher average test accuracy for sub-optimal initial choices of the window length when compared to a baseline with a fixed window length. In addition, we analyse the computational cost of DMEL and compare to a standard hyperparameter search over different window lengths, showing favorable results for DMEL. Finally, an empirical evaluation on a carefully designed dataset is performed to investigate if the differentiable spectrogram actually learns the optimal window length. The design of the dataset relies on the theory of spectrogram resolution. We also empirically evaluate the convergence rate to the optimal window length.

Proceedings: ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, South Korea, 2024.

Keywords: Deep learning, STFT, learnable Mel spectrogram, audio classification, adaptive transforms

1 Introduction

An increasing interest for using time-frequency images for feature extraction is seen in classification of audio data, typically human speech, music, and bioacoustics recordings. In audio classification, the spectrogram, the squared magnitude of the short-time Fourier transform (STFT), is typically mapped onto the Mel-scale using a Mel-filterbank [1]. This is called the Mel-spectrogram, which is then used as input to a neural network model.

The choice of Mel-filterbank affects the frequency resolution and additionally the choice of window length of the STFT creates a trade-off between time- and frequency resolution. Different trade-offs may be optimal for different audio classification tasks. Recent work has proposed the differentiable STFT (DSTFT) [2, 3, 4] where the window length, and thereby the time-frequency (TF) resolution, can be jointly optimized with the neural network. In [2] the DSTFT is proposed using a 50% fixed-overlap STFT and a Gaussian window. In [3] the constraint on the window function being Gaussian is relaxed, and a theory for a family of differentiable STFTs is presented. In these methods, the number of frequency bins is proportional to the window length, which can lead to high computational demands. Evaluations on complex audio classification tasks is therefore limited.

Recent work on learnable Mel-spectrograms include learning the Mel-filterbank [5], the energy normalization [6], and combinations of both [7, 8]. In this way, the feature extraction method can be optimized for the audio classification task at hand. Applications are seen in speech processing [9, 10], bird acoustic classification [11], and underwater acoustic classification [12].

In this work, we propose DMEL, the differentiable log-Mel spectrogram, which is an extension of DSTFT. DMEL is evaluated in a state-of-the-art convolutional neural network (CNN) for audio classification on complex audio datasets. We analyse the computational cost of DMEL and we also investigate the classification accuracy as well as the convergence rate for a simplified case. This is a step towards closing the gap between the DSTFT and the recent work using trainable filter-banks and normalization in the Mel-spectrogram for audio classification.

2 DMEL: Differentiable log-Mel spectrogram

The spectrogram is defined as

$$\begin{aligned} S_{x,\lambda}(t,f) &= |F(t,f)|^2 = \\ &= |\int_{-\infty}^{\infty} x(s-t)h(s)\exp(-i2\pi fs)ds|^2 \end{aligned} \tag{2.1}$$

where $F(t,f)$ is the short-time Fourier transform (STFT) of the signal $x(t)$ using a Gaussian window

$$h(t) = \exp(-\frac{t^2}{2\lambda^2}), \tag{2.2}$$

with scaling parameter λ which controls the window length and thereby the TF resolution of the spectrogram.

The STFT is differentiable with respect to the window parameter λ according to

$$\frac{dF(t,f)}{d\lambda} = \int_{-\infty}^{\infty} x(s-t)\frac{dh(s)}{d\lambda}\exp(-i2\pi fs)ds, \tag{2.3}$$

and a loss function $\mathcal{L}$ is differentiable w.r.t λ through gradient backpropagation using

$$\frac{d\mathcal{L}}{d\lambda} = \sum_{n=1}^{N}\sum_{k=0}^{K-1}\frac{d\mathcal{L}}{dF(n,k)}\frac{dF(n,k)}{d\lambda}, \tag{2.4}$$

where $F(t,f)$ is discretized to $F(n,k)$ with a fixed number of N bins in time and K bins in frequency [3]. Equations (2.1)-(2.4) define the differentiable spectrogram (DSPEC).

The model layer studied in this paper is the differentiable log-Mel spectrogram (DMEL), which is a novel extension where a set of Mel-filters $\{\psi_m\}_{m=1}^{M}$ are applied to DSPEC to map it to the Mel-scale

$$M_{x,\lambda}(n,m) = \log(\sum_{k=0}^{K-1} S_{x,\lambda}(n,k)\psi_m(k) + \epsilon). \tag{2.5}$$

The Mel-filters are defined as in [1] and $\epsilon = 1e^{-10}$ to avoid the logarithm of zero. The log-Mel filterbank preserves the gradients during backpropagation giving a log-Mel spectrogram with a trainable window size. We also let

$$l_\lambda = 1000 * 6 * \lambda / F_s, \tag{2.6}$$

denote the window length in milliseconds (ms), where F_s is the sample rate, and the factor 1000 converts to ms.

3 Audio data experiments

In this section we present the models and data used to evaluate DMEL, and the experiment setup. The results are then presented for each dataset and compared to the baseline. The baseline is the same model, but using a log-Mel spectrogram with a fixed window length as input layer instead of DMEL.

Table 1: The PANNs 6 layer convolutional neural network (CNN6) architecture.

Model	CNN6
Input	DMEL / baseline, 64 Mel bins
Conv. layers	(3x3 @ 64, BN, ReLU) x 2
	(3x3 @ 128, BN, ReLU) x 2
	(3x3 @ 256, BN, ReLU) x 2
	(3x3 @ 512, BN, ReLU) x 2
	Global average pooling
	FC 512, ReLU
Output	FC 50, Sigmoid

3.1 Audio data and models

DMEL is evaluated together with a linear model called "LNet" consisting of a linear layer followed by a softmax normalization, and a state-of-the-art convolutional neural network called "CNN6" from the PANNs family [13] detailed in table 1. We set the number of Mel-filters to $M = 64$ and the fixed hop length in the STFT is set to 10 ms. This imposes a bound on the achievable TF resolution, but is necessary to make the optimization feasible.

We evaluate DMEL in these two models on the audio MNIST dataset (A-MNIST) [14] and the ESC50 dataset [15]. The A-MNIST dataset consists in total of $30,000$ recordings collected from 60 different people speaking the numbers 0 to 9. The ESC50 dataset consists in total of $2,000$ recordings collected from 50 different environmental sound classes. All audio recordings are downsampled to $8,000$ Hz.

3.2 Experiments and results

The models are trained using the Adam optimizer for 100 epochs and the model with the lowest validation loss is chosen. The loss function is the standard cross-entropy loss. The task is to predict the ground truth class given the audio recording. For A-MNIST the recordings are split $60\%/20\%/20\%$ into training/validation/test datasets and for the ESC50 dataset the split is $70\%/10\%/20\%$. All parameter learning rates are 0.0001, except for λ, which is 1. The "CNN6" model was designed using a window length of 35 ms for general audio tasks [13], we therefore evaluate three different initial window lengths $l_{\lambda_{init}} \in \{10, 35, 300\}$ ms, to see if DMEL makes the model robust against this parameter choice. We do this 10 times for each model and hyper parameter configuration.

Table 2: Pairwise comparison between DMEL and the baseline for different $l_{\lambda_{init}}$ on the ESC50 dataset.

Model	$l_{\lambda_{init}}$	$l_{\lambda_{est}}$ (min, max)	Method	Accuracy
CNN6	10 ms	(25, 27) ms	**DMEL**	**87.3** $\pm$ 1.0
CNN6	10 ms	—	baseline	84.2 $\pm$ 1.2
CNN6	35 ms	(31, 90) ms	DMEL	86.1 $\pm$ 1.3
CNN6	35 ms	—	baseline	86.9 $\pm$ 0.7
CNN6	300 ms	(117, 153) ms	**DMEL**	**85.8** $\pm$ 1.2
CNN6	300 ms	—	baseline	84.7 $\pm$ 1.1

In table 2 we present the average test accuracy for the "CNN6" model on the ESC50 dataset when either using DMEL or the baseline as input layer to the model. The learned window length, denoted $l_{\lambda_{est}}$, is presented as min and max in the table. Note that the comparison is done pairwise between DMEL and the baseline for different initial window lengths, and that we do not expect DMEL to outperform the baseline when $l_{\lambda_{init}}$ is already suitable for the classification task. We use bold-face to indicate a significant difference. DMEL outperforms the baseline for the (presumably) sub-optimal choices $l_{\lambda_{init}} = 10$ ms and $l_{\lambda_{init}} = 300$ ms, and achieves similar results for $l_{\lambda_{init}} = 35$ ms (a typical choice for audio data). As a reference, the accuracy of the PANNs "CNN14" model, using a 35 ms window, is 83.3% when trained from scratch on ESC50 in the original paper [13].

Table 3: Pairwise comparison between DMEL and the baseline for different $l_{\lambda_{init}}$ on the A-MNIST dataset.

Model	$l_{\lambda_{init}}$	$l_{\lambda_{est}}$ (min, max)	Method	Accuracy
LNet	10 ms	(314, 442) ms	**DMEL**	**94.9** $\pm$ 1.0
LNet	10 ms	—	baseline	89.3 $\pm$ 1.0
LNet	35 ms	(398, 484) ms	**DMEL**	**95.0** $\pm$ 0.8
LNet	35 ms	—	baseline	91.9 $\pm$ 1.2
LNet	300 ms	(516, 608) ms	DMEL	95.3 $\pm$ 0.6
LNet	300 ms	—	baseline	95.3 $\pm$ 0.8

In table 3 we present the average test accuracy for the "LNet" model on the A-MNIST dataset for different initial window lengths. A high accuracy is achieved for surprisingly large window lengths. DMEL learns this, achieving a high accuracy for all initial window lengths, significantly outperforming the baseline for $l_{\lambda_{init}} = 10$ ms and $l_{\lambda_{init}} = 35$ ms.

The results show that DMEL makes the audio classification model more robust to the choice of the initial window length, by adapting the window length to the task at hand. We note that DMEL introduces redundancy in the TF image due to the constant hop length, and in the following section we analyse the cost of DMEL.

4 Computational cost analysis of DMEL

The complexity of a fast Fourier transform (FFT) is $\mathcal{O}(L \log L)$, where $L = 6\lambda$ is the window size in samples. In the STFT, the FFT is applied N/c times, where N is signal length and c is hop size, which results in a TF image of pixel-size $n = MN/c$. The computational complexity of a CNN is $\mathcal{O}(n)$.

As baseline, we choose $c = L/2$, avoiding redundant information in the TF image, and we assume that a hyperparameter search is done linearly between 20 ms and 300 ms over D different window sizes, thus $n = 2MN/L$ where $M = 64$ is the number of Mel-bands. Using the cost constant C_1 for the FFT and the cost constant C_2 for the CNN we can derive the following computational cost expression for the baseline

$$C_{\text{baseline}} = BC_1 \sum_{i=1}^{D} N \log L_i + BC_2 \sum_{i=1}^{D} \frac{2MN}{L_i}, \tag{4.1}$$

where L_i is the different window lengths of the hyperparameter search and $B = |l_{\lambda_{opt}} - l_{\lambda_{init}}|/\alpha$, with $\alpha = 0.001$, is the assumed steps needed until convergence to the optimal window length $l_{\lambda_{opt}} = 35$ ms. For DMEL we do not need to train D different models, but need to set the hop size $c = 80$ (10 ms) to a constant, resulting in the cost expression

$$C_{\text{DMEL}} = C_1 \sum_{i=1}^{B} \frac{NL_i}{c} \log L_i + \frac{BC_2MN}{c}. \tag{4.2}$$

The relation $C_{\text{DMEL}}/C_{\text{baseline}}$ is independent of N and is depicted for different D in figure 1. When the computational cost is dominated by the neural network, $C_2 \gg C_1$, we see computational benefits for growing D, e.g. we see that DMEL requires half the computational cost compared to baseline for $D \approx 10$. In the case when $C_1 \gg C_2$, when the cost is dominated by the FFT, we see the highest reduction for a short initial window in DMEL (blue line).

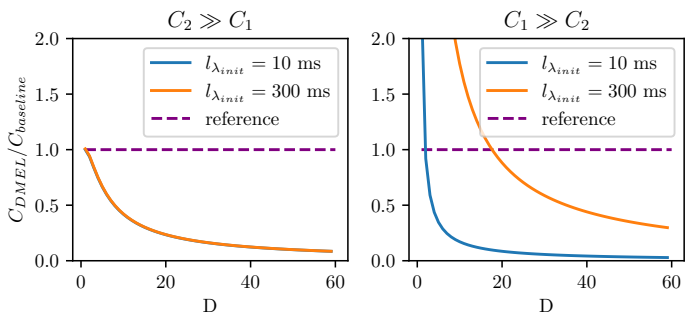


Figure 1: The computational cost quotient $C_{\text{DMEL}}/C_{\text{baseline}}$ with respect to D, for two different $l_{\lambda_{init}}$, when $C_2 \gg C_1$ (left; blue and orange lines are overlapping) and $C_1 \gg C_2$ (right).

5 Evaluation of Classification Accuracy and Convergence Rate

To investigate classification accuracy, we present a synthetic dataset, for which the accuracy should be directly dependent on the TF resolution, i.e. the window length. This is a simplified analysis and we therefore use DSPEC instead of DMEL to search for the scaling parameter. We will also verify the findings from a theoretical aspect and investigate how the convergence rate depends on the initial window length.

5.1 Simulated dataset

The simulated dataset consists of three classes, class 1, a single Gaussian-pulse, class 2 and 3, with two pulses separated either in time or frequency, as exemplified in figure 2. A Gaussian-pulse is defined as

$$g(n_0, f_0, \sigma) = A \exp(-\frac{(n-n_0)^2}{2\sigma^2}) \sin(2\pi f_0 n + \phi), \tag{5.1}$$

where the parameters are chosen to give TF symmetry ($\sigma = 6.4$) and optimal TF separation for an optimal window length ($\lambda = 6.4$), see section 5.3. Gaussian noise is added to all signal classes and A, ϕ, n_0 and f_0 are chosen at random. Full parameter description is given in source code (page 1). In total $5,000$ samples is used, with approximately $1/3$ of each class. Using $N = 128$ with hop size one and $K = 256$, the resulting TF images are of square size $N \times K/2$.

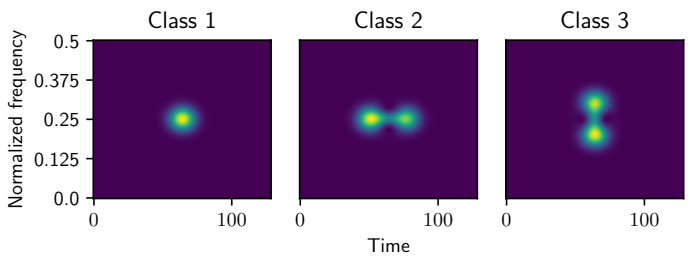


Figure 2: Examples from the simulated Gaussian-pulse dataset.

5.2 Experiment and results

We evaluate "LNet" on this dataset using either DSPEC or a fixed window length baseline, which is trained in the same way as the audio classification model, with the exception that Stochastic Gradient Descent (SGD) is used.

The λ_{init} which on average give the highest test accuracy for both methods is $\lambda_{init} = \sigma = 6.4$ (see table 4). DSPEC is able to learn a λ_{est} close to 6.4 for all λ_{init}, significantly outperforming the baseline for the sub-optimal choices.

Table 4: Pairwise comparison between DSPEC and the baseline for different $l_{\lambda_{init}}$ on the Gaussian-pulse dataset.

Model	λ_{init}	λ_{est} (min, max)	Method	Accuracy
LNet	1.3	(4.8, 5.6)	**DSPEC**	**98.5** $\pm$ 0.2
LNet	1.3	—	baseline	95.5 $\pm$ 0.2
LNet	6.4	(5.3, 6.0)	DSPEC	98.5 $\pm$ 0.2
LNet	6.4	—	baseline	98.8 $\pm$ 0.3
LNet	31.9	(3.4, 6.5)	**DSPEC**	**98.0** $\pm$ 0.4
LNet	31.9	—	baseline	94.9 $\pm$ 0.4

5.3 Theory on time-frequency resolution and symmetry

In this subsection we derive the parameter choices for optimal TF resolution and TF symmetry. The reasoning relies on that the resolution limit of two closely spaced Gaussian functions is two times the actual Gaussian function scaling parameter, [16]. The window is Gaussian with parameter λ and the signal is $g(0, 0, \sigma)$, which can be generalized to any n_0, f_0 due to TF shift invariance [17]. The resulting discretized spectrogram is

$$S_x(n, k) = \frac{2\lambda^2\sigma^2\pi}{\lambda^2 + \sigma^2} \exp(-\frac{1}{2}(\frac{n}{\delta_t})^2 - \frac{1}{2}(\frac{k}{\delta_f})^2), \tag{5.2}$$

a two-dimensional Gaussian function with scaling parameters

$$\delta_t = \sqrt{\frac{\lambda^2 + \sigma^2}{2}}, \quad \delta_f = \frac{K}{2\pi\lambda\sigma}\sqrt{\frac{\lambda^2 + \sigma^2}{2}}. \tag{5.3}$$

Minimizing δ_t and δ_f will result in optimal TF resolution. The corresponding TF cross-section area is $A = \pi\delta_t\delta_f$ with derivative as

$$\frac{dA}{d\lambda} = \frac{K\sigma}{4}(\frac{1}{\sigma^2} - \frac{1}{\lambda^2}), \tag{5.4}$$

giving $\lambda_{opt} = \sigma$ and $A_{min} = K/2 = 128$. Optimal TF resolution is therefore given using a matched window [17]. For TF symmetry (equal number of bins in time- and frequency) we also set $\delta_t = \delta_f$ in (11), and we find for the matched window case, $\lambda_{opt} = \sigma = \sqrt{K/(2\pi)} = 6.4$.

5.4 Convergence rate

Relying on TF symmetry we perform an experiment on the difference in convergence rate for λ when approaching the optimal solution from a small initial value, or a large initial

Table 5: Convergence rate experiment

λ_{init}	1.3	31.9
Iterations	75.5 ± 1.3	186.0 ± 37.8

value of λ. We compute the optimally concentrated spectrogram for a Gaussian-pulse signal and use this as the ground truth (see leftmost image in figure 2). The task is to learn this spectrogram (i.e., the true value λ_{opt}) using DSPEC with SGD and a mean-squared error (MSE) loss. The MSE loss is between the estimated spectrogram and the ground truth. We study two carefully chosen values for λ_{init}. Both give exactly the same cross-section area A and therefore also the same initial MSE loss. We then measure the number of SGD iterations until $|\lambda_{est} - \lambda_{opt}| < 0.1$, and present the average number of iterations until convergence over 50 signals in table 5. The convergence rate is twice as fast when λ_{init} is chosen as the value smaller than λ_{opt}. Faster convergence means a smaller B in (4.2), leading to further reduction in computational cost.

6 Conclusions

We introduce DMEL: a differentiable log-Mel spectrogram for audio classification, allowing joint optimization of the window length and the neural network. DMEL achieves a higher test accuracy on average than the baseline with a fixed window length for all non-optimal initial window lengths on all evaluated datasets. In addition, we show that DMEL leads to a reduced computational cost compared to a standard hyperparameter search over different window lengths, especially if the initial window length is short. An empirical evaluation shows that the differentiable spectrogram is able to learn the optimal window length on a carefully designed classification task. Finally, a convergence rate experiment indicates that a shorter window is beneficial for fast convergence. The overall results suggest that it is favorable to initialize the DMEL with a short window, resulting in lower computational cost and faster convergence.

References

[1] Malcolm Slaney. Auditory toolbox. Technical report, Interval Research Corporation, 1998.

[2] An Zhao, Krishna Subramani, and Paris Smaragdis. Optimizing short-time Fourier transform parameters via gradient descent. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2021-June(2):736–740, 2021. ISSN 15206149. doi: 10.1109/ICASSP39728.2021.9413704.

[3] Maxime Leiber, Axel Barrau, Yosra Marnissi, and Dany Abboud. A differentiable short-time Fourier transform with respect to the window length. In *2022 30th European Signal Processing Conference (EUSIPCO)*, pages 1392–1396, 2022. doi: 10.23919/EUSIPCO55093.2022.9909963.

[4] Maxime Leiber, Yosra Marnissi, Axel Barrau, and Mohammed El Badaoui. Differentiable Adaptive Short-Time Fourier Transform with Respect to the Window Length. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10095245.

[5] Neil Zeghidour, Nicolas Usunier, Iasonas Kokkinos, Thomas Schaiz, Gabriel Synnaeve, and Emmanuel Dupoux. Learning Filterbanks from Raw Speech for Phone Recognition. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2018-April:5509–5513, 2018. ISSN 15206149. doi: 10.1109/ICASSP.2018.8462015.

[6] Yuxuan Wang, Pascal Getreuer, Thad Hughes, Richard F. Lyon, and Rif A. Saurous. Trainable frontend for robust and far-field keyword spotting. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, (1):5670–5674, 2017. ISSN 15206149. doi: 10.1109/ICASSP.2017.7953242.

[7] Neil Zeghidour, Olivier Teboul, Félix de Chaumont Quitry, and Marco Tagliasacchi. LEAF: A Learnable Frontend for Audio Classification. In *ICLR, International Conference on Learning Representations*, pages 1–16, 2021. URL `http://arxiv.org/abs/2101.08596`.

[8] Jan Schlüter and Gerald Gutenbrunner. EfficientLEAF: A Faster LEarnable Audio Frontend of Questionable Use. *European Signal Processing Conference*, 2022-Augus: 205–208, 2022. ISSN 22195491.

[9] Miguel Arjona Ramirez, Wesley Beccaro, Demostenes Zegarra Rodriguez, and Renata Lopes Rosa. Differentiable Measures for Speech Spectral Modeling. *IEEE Access*, 10:17609–17618, 2022. ISSN 21693536. doi: 10.1109/ACCESS.2022.3150728.

[10] Quchen Fu, Zhongwei Teng, Jules White, Maria E. Powell, and Douglas C. Schmidt. Fastaudio: a Learnable Audio Front-End for Spoof Speech Detection. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2022-May:3693–3697, 2022. ISSN 15206149. doi: 10.1109/ICASSP43922.2022.9746722.

[11] Mark Anderson and Naomi Harte. Learnable Acoustic Frontends in Bird Activity Detection. In *International Workshop on Acoustic Signal Enhancement, IWAENC 2022 - Proceedings*, 2022. ISBN 9781665468671. doi: 10.1109/IWAENC53105.2022.9914694. URL `http://arxiv.org/abs/2210.00889`.

[12] Jiawei Ren, Yuan Xie, Xiaowei Zhang, and Ji Xu. UALF: A learnable front-end for intelligent underwater acoustic classification system. *Ocean Engineering*, 264 (September), 2022. ISSN 00298018. doi: 10.1016/j.oceaneng.2022.112394.

[13] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 28:2880–2894, nov 2020. ISSN 2329-9290. doi: 10.1109/TASLP.2020.3030497. URL `https://doi.org/10.1109/TASLP.2020.3030497`.

[14] Sören Becker, Marcel Ackermann, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Interpreting and explaining deep neural networks for classification of audio signals. *CoRR*, abs/1807.03418, 2018. URL `http://arxiv.org/abs/1807.03418`.

[15] Karol J. Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, page 1015–1018, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450334594. doi: 10.1145/2733373.2806390. URL `https://doi.org/10.1145/2733373.2806390`.

[16] Hajo Holzmann and Sebastian Vollmer. A likelihood ratio test for bimodality in two-component mixtures with application to regional income distribution in the EU. *AStA Advances in Statistical Analysis: A Journal of the German Statistical Society*, 92(1):57 – 69, 2008. ISSN 1863-8171.

[17] Leon Cohen. *Time-Frequency Analysis*. Prentice-Hall, 1995.

Paper D

Few-shot bioacoustic event detection using an event-length adapted ensemble of prototypical networks

John Martinsson[1,2], Martin Willbo[1], Aleksis Pirinen[1], Olof Mogren[1], Maria Sandsten[2]

[1] *Computer Science, RISE Research Institutes of Sweden*
[2] *Centre for Mathematical Sciences, Lund University*

Abstract

In this paper we study two major challenges in few-shot bioacoustic event detection: variable event lengths and false-positives. We use prototypical networks where the embedding function is trained using a multi-label sound event detection model instead of using episodic training as the proxy task on the provided training dataset. This is motivated by polyphonic sound events being present in the base training data. We propose a method to choose the embedding function based on the average event length of the few-shot examples and show that this makes the method more robust towards variable event lengths. Further, we show that an ensemble of prototypical neural networks trained on different training and validation splits of time-frequency images with different loudness normalizations reduces false-positives. In addition, we present an analysis on the effect that the studied loudness normalization techniques have on the performance of the prototypical network ensemble. Overall, per-channel energy normalization (PCEN) outperforms the standard log transform for this task. The method uses no data augmentation and no external data. The proposed approach achieves a F-score of 48.0% when evaluated on the hidden test set of the Detection and Classification of Acoustic Scenes and Events (DCASE) task 5.

Proceedings: In proceedings of the 2022 Detection and Classification of Acoustic Scenes and Events (DCASE) Workshop, Nancy, France, 2022.

Keywords: Machine listening, bioacoustics, few-shot learning, ensemble

1 Introduction

The human-induced accelerated loss in biodiversity [1] has led to a need for automated and low-cost wildlife monitoring where machine learning is a promising way forward [2]. Passive acoustic monitoring (PAM) is becoming an important tool in ecology for monitoring animal populations through their vocalizations [3]. Annotating PAM data is costly and requires specific domain expertise which motivates research on few-shot learning for bioacoustic event detection [4]. The goal of few-shot bioacoustic event detection is to detect the onset and offset of animal vocalizations in sound recordings using only a few annotated examples.

Recent work has demonstrated that prototypical networks are a promising approach for few-shot sound event detection [5, 6, 7], but a remaining challenge is high variance in classification accuracy between models because of the small amount of training data. Recent work on audio classification and sound event detection has demonstrated promising results using ensembles [8, 9, 10]. Ensembles may be especially useful for the few-shot task due to the high variance in classification accuracy between models [11]. To the best of our knowledge, prior work on ensemble methods for few-shot sound event detection remains understudied and motivated by this we study the effect of using an ensemble of prototypical networks for few-shot bioacoustic event detection.

Another challenge in few-shot sound event detection is the high variability in event lengths for the different event classes [6]. The event lengths can range from milliseconds to multiple seconds, which necessitates methods capable of adapting to the task specific event lengths. Wang et al. [6] suggest that future work should look into adapting the context window to the few-shot task. A common approach is to use a model which can handle variable context windows, train using a fixed context window, and at test time adapt the context window to the few-shot task. In this work we propose choosing the embedding function as well as the context window based on the few-shot examples. The embedding function is chosen from a set of embedding functions trained on different context windows, thus acting as experts on certain event lengths. Another way to approach this problem is by using a proposal based method [12].

2 Method

In this section we present our method which is based on prototypical networks [13] and extended with an event-length adapted ensemble. We describe how each embedding function for the prototypical networks is trained and how the embedding functions are selected based on the few-shot examples to produce an ensemble prediction at test time. The full source code and instructions on how to reproduce the results can be found at:

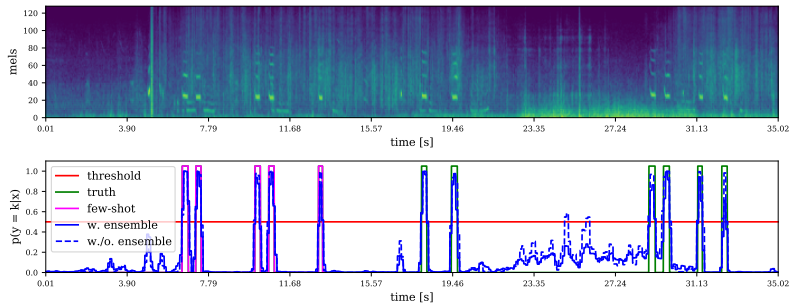


Figure 1: A log Mel spectrogram of part of a sound recording (top) and examples of predictions (bottom) from an ensemble prototypical network (solid blue line) and a prototypical network (dashed blue line) as well as the given few-shot examples (purple line) and remaining ground truth events (green line). The decision threshold τ is 0.5 (red line).

`https://github.com/johnmartinsson/few-shot-learning-bioacoustics`.

2.1 Training the embedding function

The goal is to learn an embedding function from the base training data, acting as a proxy task for the few-shot task. The base training data set consists of annotated sound recordings for 47 known event classes and one "unknown" event class. The set of sound event classes are disjoint between the base training data and the few-shot task. We are given the start and end times $\mathcal{A}_k = \{(s_i^k, e_i^k)\}_{i=1}^N$ of these classes, where (s_i^k, e_i^k) denotes the start and end time of sound event class k for annotation i. There is overlap in the annotations, i.e. two different sound events can occur (partially) simultaneously, and we therefore treat this as a multi-label problem. We model the 47 known sound event classes and the "unknown" sound event class identically, yielding a total of $K = 48$ classes.

We assume that a fixed length audio segment $x \in \mathbb{R}^T$, consisting of T consecutive audio samples, is fed to the embedding function $f_\theta^T : \mathbb{R}^T \to \mathbb{R}^M$ (see section 2.4 for further details), where $M \ll T$. We split the audio recordings into audio segments $x_i \in \mathbb{R}^T$ by sliding a window of size T with a hop size of $T/2$ over each recording. For each audio segment x_i, a target vector $y_i \in \{0, 1\}^{K \times n}$ is derived. If $n = T$ it means that the target contains one label per audio sample. Choosing $n < T$ means that the temporal resolution for the target is reduced. The resulting dataset $\mathcal{D}_b = \{(x_i, y_i)\}_{i=1}^N$ defines the sound event detection task used to train the embedding function.

A prediction of the target classes for a given audio segment x_i is derived by $\hat{y}_i = h_\phi(f_\theta^T(x_i))$, where $h_\phi(\cdot)$ is a linear layer followed by an element-wise sigmoid activation function, and $f_\theta^T(\cdot)$ is a convolutional neural network where the first layer is a (non-learnable) time-frequency transform.

The loss function is the mean element-wise binary cross-entropy between the target y_i and the prediction $\hat{y}_i$, where the mean is taken over the class dimension K and the temporal dimension n.

For a fixed T, we train a set of C different embedding functions, parameterized as $\Theta = \{\theta_1, \ldots, \theta_C\}$, each with different randomly initialized weights of the neural network, different training and validation splits of the base training data, and different time-frequency transforms in the first layer of the embedding function.

2.2 Prototypical network at test time

At test time we are given a sound recording and the $M = 5$ first event examples of the class of interest. We denote these $A_p = \{(s_i, e_i)\}_{i=1}^{M}$ and call them the *positive* sound events. We assume that the gaps between the positive event annotations are background noise and let $A_n = \{(e_i, s_{i+1})\}_{i=1}^{M-1}$ denote the start and end time of the $M-1$ first *negative* sound events. We assume the likelihood of an annotator missing events to be low.

Let $l_i = e_i - s_i$ be the length of annotation i. If $l_i < T$ we "expand" the annotation with the $(T - l_i)/2$ preceding and subsequent audio samples to get an audio segment of length T, and if $l_i \geq T$ we do not expand. We then split this into segments of length T by sliding a window of size T over the signal with a hop size of $T/16$ (if expanded this will only result in one segment). Let S_p denote the set of positive audio segments derived from these annotated start and end times, and let S_n denote the set of negative audio segments. We use the embedding function f_θ^T and define the prototypes as

$$c_k = \frac{1}{|S_k|} \sum_{x \in S_k} f_\theta^T(x) \tag{2.1}$$

and derive a pseudo-probability of audio segment x belonging to sound class k from the prototypical network by

$$p_\theta(y = k|x) = \frac{\exp(-d(f_\theta^T(x), c_k))}{\sum_{k'} \exp(-d(f_\theta^T(x), c_{k'}))}, \tag{2.2}$$

where $k \in \{n, p\}$ and $d(f_\theta^T(x), c_k)$ denotes the Euclidean distance between the query $f_\theta^T(x)$ and the prototype c_k.

The query set S_q is derived by sliding a window of size T over the signal with a hop size of $T/2$. The reason for setting the hop size relative to T is that this means that we do equally many predictions for each audio sample in the validation recordings.

2.3 Our contributions

We now present the two main contributions of this paper: i) an event-length adapted embedding function for the few-shot task, and ii) using an ensemble of predictions.

Adapting the embedding function. We use the annotated positive events $A_p = \{(s_i, e_i)\}_{i=1}^{M}$ and compute the set of event lengths $L = \{e_i - s_i\}_{i=1}^{M}$. We choose $T^* \in \{T_1, 2^1 T_1, 2^2 T_1, 2^3 T_1\}$ such that $\sqrt{(T - l_{\text{min}}/2)^2}$ is minimized, where l_{min} is the shortest event length in L.

We choose $T_1 = 2048$ which is 0.09 seconds at a sampling rate of 22050 Hz so that we can plausibly detect the shortest events in the few-shot validation set. We limit the amount of extra computation needed during training and the extra memory needed during inference by setting the maximum T to $2^3 T_1$.

Ensemble. Let $\Theta = \{\theta_i^{T^*}\}_{i=1}^{C}$ denote the set of parameters of C different prototypical network models adapted to the average event length of the few-shot task. Then we define

$$p_\Theta(y = k|x) = \frac{1}{C} \sum_{\theta \in \Theta} p_\theta(y = k|x) \tag{2.3}$$

as in [14], which can be viewed as a uniformly-weighted mixture of experts. We say that x belongs to the positive event class if $p_\Theta(y = p|x) > \tau$ and otherwise x belongs to the negative event class. This is done for every $x \in S_q$. Finally, if the query is classified as a positive event then the start and end time associated with that query is used as the predicted positive event timings.

2.4 Details of the embedding function

The embedding function consists of a time-frequency transform followed by a convolutional neural network, both of which are briefly described below.

Time-frequency transform. The first layer of the embedding function is a time-frequency transform. We use the Mel transform where the number of Mel bins is 128, the window size is roughly 25ms, and the hop size is half the window size. We either use the log transform as a loudness normalization or we use PCEN [15] with fixed parameters developed for speech audio or for bioacoustics as suggested in [16].

Convolutional neural network. The convolutional neural network used is an adapted version of the 10-layer residual neural network [17] used in the baseline for the challenge. Specifically, we i) add the classification head $h_\phi(\cdot)$ so that we can model the defined multi-

label task, ii) use the same number of filters in every convolutional layer, and iii) reduce the max pooling along the time-dimension when audio segments are too short.

2.5 Evaluation metric

The method is evaluated by taking the harmonic mean over the F-scores for the different subsets in the evaluation sets. The F-score is computed by a bi-partite matching between the predicted and ground truth events, where the requirement for a match is an intersection-over-union (IoU) of at least 0.3 [4].

2.6 Post-processing

Since we get one prediction for each query audio segment, this limits the possible length of the prediction with this model. To solve this, we simply merge all overlapping predicted positive events into one detected event with a single start and end time.

A predicted positive event will only be considered to be a match with a true positive event during evaluation if they have an intersection-over-union (IoU) of at least 0.3. We therefore remove predictions which are shorter than $0.3 * l_{\text{avg}}$ or longer than $(1/0.3) * l_{\text{avg}}$, where l_{avg} is the average event length of the given five annotations. Since predictions of these lengths can on average not be matched with true events as the evaluation is defined.

3 Data

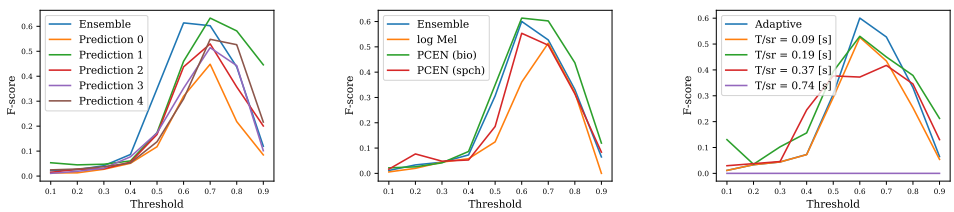
We use the few-shot examples to compute the mean event length, the mean gap length, and the density of annotated sound events – see table 1. The few-shot validation set consists of three different subsets: HB, ME, and PB. The HB subset contains long events with low noise. The ME subset contains short events with low noise. The PB subset contains very short events with very high noise. The mean event length is defined as the mean length of the five annotated events; the mean gap length is defined as the mean length of the *unannotated* gaps between the five annotated events; and the density is the sum of the time of the five annotated events divided by the total time. A full description of the dataset can be found in [18].

Table 1: Few-shot validation data statistics.

Subset	Mean event length	Mean gap length	Mean density
HB	11.25 ± 3.11	6.12 ± 5.39	0.73 ± 0.12
ME	0.22 ± 0.03	1.40 ± 0.04	0.17 ± 0.02
PB	0.12 ± 0.08	59.89 ± 55.55	0.01 ± 0.02

4 Experiments and Results

We have trained each embedding function on the described multi-label task on the base training data using the Adam [19] optimizer with a learning rate of $1e{-}3$. The network is trained on a random split with 80% training data and validated on the remaining 20%. Each network in the ensemble is trained on a different random split. The training proceeds until we have observed no reduction in validation loss for the last 10 epochs and the model with the lowest validation loss is chosen as the final model. The temporal resolution of the targets have been fixed to $n = 16$, meaning that we have 16 targets for any given audio segment.



(a) Ensemble over seeds, and training splits.

(b) Ensemble over time-frequency transforms.

(c) Adapted embedding function.

Figure 2: A comparison between: (a) an ensemble of five predictions using embedding functions trained on PCEN (bioacoustics) features with each of the individual predictions, (b) an ensemble of embedding functions trained and tested on log Mel, PCEN (bioacoustics), or PCEN (speech), with an ensemble of predictions over all three, and (c) the adaptive embedding function with using each of the fixed size embedding functions respectively (sr denotes the sample rate). All results in the figure are derived on the few-shot validation set.

In figure 2a we compare the F-score achieved on the few-shot validation set when using an ensemble of five predictions with using each of these predictions by themselves. The time-frequency transform used is PCEN (bioacoustics). The achieved F-score by the ensemble is higher than the best of these individual predictions for $0.4 \leq \tau \leq 0.6$, and outperforms or matches the mean of them for other τ. We also note that the optimal τ is around 0.7 for the single predictions, and moves to 0.6 for the ensemble.

In figure 2b we compare the F-score of a five prediction ensemble for each time-frequency transform and compare this to an ensemble over all three time-frequency transform ensembles. We do not observe a significant increase in F-score when comparing the time-

frequency ensemble to the ensemble using the PCEN (bioacoustic) time-frequency transform, but the time-frequency ensemble outperforms the ensemble using PCEN (speech) and log Mel transform. The optimal threshold τ varies around 0.6 to 0.7 for the ensembles using a single transform, and is at 0.6 for the time-frequency ensemble.

In figure 2c we compare the F-score achieved on the few-shot validation set when using the event-length adapted embedding functions in the ensemble with using any of the fixed $T \in \{T_1, 2^1 T_1, 2^2 T_1, 2^3 T_1\}$. Adapting the embedding function increases performance from 53.0% (using best $T = 4096$) to 60.0% F-score for $\tau = 0.6$.

In table 2 we show an ablation study. Adapting the embedding function increases the F-score on average with 8.3 percentage points, and adding the ensemble increases the F-score an additional 11.4 percentage points. We compare against a prototypical network using an embedding function (no ensemble) which has been trained on PCEN (speech) and a best performing fixed segment length of 4096. The F-score when no ensemble is performed is the average (and standard deviation) over each single network in the ensemble.

Table 2: An ablation study of our system on the few-shot validation set where we add adaptive embedding functions and ensemble.

Method	Ensemble	Adaptive	F-score
Ours	No	No	41.3 ± 3.8
Ours	No	Yes	49.6 ± 5.3
Ours	Yes	Yes	60.0

In table 3 we present the F-score from the final evaluation on the hidden test set from the challenge. We include information on whether or not the system uses data augmentation techniques or external datasets during training.

Table 3: The final F-score evaluation on the hidden test set for the baselines provided by the challenge organizers: template matching (TM) and prototypical networks (PN), and the top five submissions for the challenge.

System	External data	Augmentation	F-score
Baseline (TM)	No	No	12.3
Baseline (PN)	No	No	5.3
Ours [20]	No	No	48.0
Tang et al., [21]	No	No	62.1
Liu et al., [22]	Yes	Yes	48.2
Hertkorn [23]	No	No	44.4
Liu et al., [24]	Yes	Yes	44.3

5 Discussion and Conclusions

In this section we will discuss our results and relate them to the baselines which were all developed concurrently to our work.

During development of this method we observed that random sampling of S_n, the set of negative examples does not work well for validation files with high event densities, which is why we chose to use the gaps between the first five annotated events instead. This observation was also made in concurrent work submitted to the challenge [21, 22, 24]

We further observed that a fixed audio segment size T resulted in poor predictive performance on the few-shot validation set in cases where event-lengths deviated from size T. Indicating the importance of adapting the embedding function.

We observed that the optimal threshold was different for the few-shot validation tasks and choosing a default value of $\tau = 0.5$ to be detrimental. However, finding an optimal threshold for the few-shot tasks is a difficult problem. Using an ensemble alleviates this issue by moving the optimal threshold closer to the default value.

The ensemble improves performance by correctly predicting most true positives, while reducing the number of false positives. This could intuitively be thought of as the ensemble being in agreement for true positive predictions, the average of which still yields a high pseudo-probability, while being in disagreement when predicting false positives, the average of which would be closer to 0.5. This effect can be seen in figure 1, where some of the false positives predicted when not using an ensemble (dashed blue line) are removed by using an ensemble of the predictions (solid blue line), leading to a reduction in false-positives.

The baselines in this study were all developed concurrently to our work. Tang et al., [21] propose using a frame-level cross-entropy loss function for training instead of episodic training as the proxy task. Our approach is similar when setting the temporal resolution n of the target vector to the number of frames in the time-frequency image. The effect of varying temporal resolutions n for the proxy task would be interesting to study in future work. Tang et al. [21] further propose an iterative training scheme to adapt their method to the few-shot task [21] where the unlabeled audio in the test files is iteratively classified and then used for training. Liu et al. [22] and Liu et al. [24] use transductive inference to better adapt to the evaluation set, and Hertkorn [23] studies the importance of choosing appropriate parameters for the used time-frequency transform.

In conclusion, we have shown that choosing the embedding function based on the event lengths will increase performance, and that false-positives can be reduced by an ensemble of predictions. We have also shown that out of the three time-frequency transforms we have studied, PCEN (bioacoustics) performs best, followed by PCEN (speech) and log Mel.

References

[1] Sandra Díaz, Josef Settele, Eduardo S. Brondízio, Hien T. Ngo, John Agard, Almut Arneth, Patricia Balvanera, Kate A. Brauman, Stuart H.M. Butchart, Kai M.A. Chan, A. G. Lucas, Kazuhito Ichii, Jianguo Liu, Suneetha M. Subramanian, Guy F. Midgley, Patricia Miloslavich, Zsolt Molnár, David Obura, Alexander Pfaff, Stephen Polasky, Andy Purvis, Jona Razzaque, Belinda Reyers, Rinku Roy Chowdhury, Yunne Jai Shin, Ingrid Visseren-Hamakers, Katherine J. Willis, and Cynthia N. Zayas. Pervasive human-driven decline of life on Earth points to the need for transformative change. *Science*, 366(6471), 2019. ISSN 10959203. doi: 10.1126/science.aax3100.

[2] Devis Tuia, Benjamin Kellenberger, Sara Beery, Blair R. Costelloe, Silvia Zuffi, Benjamin Risse, Alexander Mathis, Mackenzie W. Mathis, Frank van Langevelde, Tilo Burghardt, Roland Kays, Holger Klinck, Martin Wikelski, Iain D. Couzin, Grant van Horn, Margaret C. Crofoot, Charles V. Stewart, and Tanya Berger-Wolf. Perspectives in machine learning for wildlife conservation. *Nature Communications*, 13 (1):1–15, 2022. ISSN 20411723. doi: 10.1038/s41467-022-27980-y.

[3] Rory Gibb, Ella Browning, Paul Glover-Kapfer, and Kate E. Jones. Emerging opportunities and challenges for passive acoustics in ecological assessment and monitoring. *Methods in Ecology and Evolution*, 10(2):169–185, 2019. ISSN 2041210X. doi: 10.1111/2041-210X.13101.

[4] Veronica Morfi, Ines Nolasco, Vincent Lostanlen, Shubhr Singh, Ariana Strandburg-Peshkin, Lisa Gill, Hanna Pamuła, David Benvent, and Dan Stowell. Few-Shot Bioacoustic Event Detection: A New Task at the DCASE 2021 Challenge. *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, (November):145–149, 2021.

[5] Bowen Shi, Ming Sun, Krishna C. Puvvada, Chieh Chi Kao, Spyros Matsoukas, and Chao Wang. Few-Shot Acoustic Event Detection Via Meta Learning. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2020-May:76–80, 2020. ISSN 15206149. doi: 10.1109/ICASSP40776.2020.9053336.

[6] Yu Wang, Justin Salamon, Nicholas J. Bryan, and Juan Pablo Bello. Few-shot sound event detection. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2020-May:81–85, 2020. ISSN 15206149. doi: 10.1109/ICASSP40776.2020.9054708.

[7] Jordi Pons, Joan Serra, and Xavier Serra. Training Neural Audio Classifiers with Few Data. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2019-May:16–20, 2019. ISSN 15206149. doi: 10.1109/ICASSP.2019.8682591.

[8] Donmoon Lee, Subin Lee, Yoonchang Han, and Kyogu Lee. Ensemble of Convolutional Neural Networks for Weakly-Supervised Sound Event Detection Using Multiple Scale Input. *DCASE 2017*, 1(November):14–18, 2017. URL http://www.cs.tut.fi/sgn/arg/dcase2017/documents/challenge_technical_reports/DCASE2017_Lee_199.pdf.

[9] Loris Nanni, Yandre M.G. Costa, Rafael L. Aguiar, Rafael B. Mangolin, Sheryl Brahnam, and Carlos N. Silla. Ensemble of convolutional neural networks to improve animal audio classification. *Eurasip Journal on Audio, Speech, and Music Processing*, 2020(1), 2020. ISSN 16874722. doi: 10.1186/s13636-020-00175-3.

[10] Loris Nanni, Gianluca Maguolo, Sheryl Brahnam, and Michelangelo Paci. An ensemble of convolutional neural networks for audio classification. *Applied Sciences (Switzerland)*, 11(13):1–27, 2021. ISSN 20763417. doi: 10.3390/app11135796.

[11] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Diversity with cooperation: Ensemble methods for few-shot classification. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-October:3722–3730, 2019. ISSN 15505499. doi: 10.1109/ICCV.2019.00382.

[12] Piper Wolters, Chris Daw, Brian Hutchinson, and Lauren Phillips. Proposal-based Few-shot Sound Event Detection for Speech and Environmental Sounds with Perceivers. pages 1–7, 2021. URL http://arxiv.org/abs/2107.13616.

[13] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, 2017-Decem:4078–4088, 2017. ISSN 10495258.

[14] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 2017-December:6403–6414, 2017. ISSN 10495258.

[15] Yuxuan Wang, Pascal Getreuer, Thad Hughes, Richard F. Lyon, and Rif A. Saurous. Trainable frontend for robust and far-field keyword spotting. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, (1):5670–5674, 2017. ISSN 15206149. doi: 10.1109/ICASSP.2017.7953242.

[16] Vincent Lostanlen, Justin Salamon, Mark Cartwright, Brian Mcfee, Andrew Farnsworth, Steve Kelling, and Juan Pablo Bello. Per-Channel Energy Normalization: Why and How. *IEEE SIGNAL PROCESSING LETTERS*, (September):1–6, 2018. URL http://www.justinsalamon.com/uploads/4/3/9/4/4394963/lostanlen_pcen_spl2018.pdf.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *Arxiv.Org*, 7(3):171–180, 2015. ISSN 1664-1078. doi: 10.3389/fpsyg.2013.00124. URL http://arxiv.org/pdf/1512.03385v1.pdf.

[18] I. Nolasco, S. Singh, E. Vidana-Villa, E. Grout, J. Morford, M. Emmerson, F. Jensens, H. Whitehead, I. Kiskin, A. Strandburg-Peshkin, L. Gill, H. Pamula, V. Lostanlen, V. Morfi, and D. Stowell. Few-shot bioacoustic event detection at the dcase 2022 challenge, 2022. URL https://arxiv.org/abs/2207.07911.

[19] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. pages 1–15, 2014. ISSN 09252312. doi: http://doi.acm.org.ezproxy.lib.ucf.edu/10.1145/1830483.1830503. URL http://arxiv.org/abs/1412.6980.

[20] John Martinsson, Martin Willbo, Aleksis Pirinen, Olof Mogren, and Maria Sandsten. Few-shot bioacoustic event detection using a prototypical network ensemble with adaptive embedding functions. Technical report, 2022.

[21] Jigang Tang, Xueyang Zhang, Tian Gao, Diyuan Liu, Xin Fang, Jia Pan, Qing Wang, Jun Du, Kele Xu, and Qinghua Pan. Few-shot embedding learning and event filtering for bioacoustic event detection. Technical report, iFLYTEK Research Institute, Hefei, China, 2022.

[22] Haohe Liu, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Wenwu Wang, and Mark D Plumbley. Surrey system for DCASE 2022 task 5 : few-shot bioacosutic event detection with segment-level metric learning. Technical report, University of Surrey, Surrey, United Kingdom, 2022.

[23] Michael Hertkorn. Few-shot bioacoustic event detection : don't waste information. Technical report, ZF Friedrichshafen AG, Friedrichshafen, Germany, 2022.

[24] Miao Liu, Jianqian Zhang, Lizhong Wang, Jiawei Peng, and Chenguang Hu. Bit SRCB teams 's submission for DCASE2022 task5 - few-shot bioacoustic event detection. Technical report, Beijing Institute of Technology, Beijing, China, 2022.