How Much Attention Should we Pay to Mosquitoes?

How Much Attention Should we Pay to Mosquitoes?

Moreno La Quatra*
moreno.laquatra@polito.it
Politecnico di Torino
Turin, Italy

Lorenzo Vaiani*
lorenzo.vaiani@polito.it
Politecnico di Torino
Turin, Italy

Alkis Koudounas*
alkis.koudounas@polito.it
Politecnico di Torino
Turin, Italy

Luca Cagliero
luca.cagliero@polito.it
Politecnico di Torino
Turin, Italy

Paolo Garza
paolo.garza@polito.it
Politecnico di Torino
Turin, Italy

Elena Baralis
elena.baralis@polito.it
Politecnico di Torino
Turin, Italy

ABSTRACT

Mosquitoes are a major global health problem. They are responsible for the transmission of diseases and can have a large impact on local economies. Monitoring mosquitoes is therefore helpful in preventing the outbreak of mosquito-borne diseases. In this paper, we propose a novel data-driven approach that leverages Transformer-based models for the identification of mosquitoes in audio recordings. The task aims at detecting the time intervals corresponding to the acoustic mosquito events in an audio signal. We formulate the problem as a sequence tagging task and train a Transformer-based model using a real-world dataset collecting mosquito recordings. By leveraging the sequential nature of mosquito recordings, we formulate the training objective so that the input recordings do not require fine-grained annotations. We show that our approach is able to outperform baseline methods using standard evaluation metrics, albeit suffering from unexpectedly high false negatives detection rates. In view of the achieved results, we propose future directions for the design of more effective mosquito detection models.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; **Supervised learning**.

KEYWORDS

Mosquito detection, Audio event detection, Transformer models, Audio sequence modelling

1 INTRODUCTION

The identification of mosquitoes is of great importance for monitoring the risk of mosquito-borne diseases and for the development of mosquito control strategies. However, recognizing the mosquito buzzing sounds is a challenging task. Aside from the difficulty of detecting faint sounds, they can be also easily confused with similar background or buzzing sounds. In recent years, data-driven methodologies have shown state-of-the-art performance on Acoustic Event Detection (AED). Deep neural networks are capable of learning complex representations of raw audio data, which can be particularly beneficial to tackle AED. However, most of the aforesaid approaches are designed for the audio classification tasks, e.g., urban sound classification [13]. Conversely, the identification of mosquitoes events is a frame-based event detection task, which requires the correct identification of intermittent and variable-length buzzing sounds.

The most common approach to identify acoustic events in long-lasting recordings is to train a deep learning model to classify short audio segments. In [8, 9] each audio frame is processed independently, and the classification results are combined to make global decisions on the whole audio recording. As a drawback, the aforesaid approaches ignore the temporal dependencies among frames, which can be crucial for the identification of variable-length mosquitoes buzzing sounds.

To address the above issue, we propose a methodology that leverages transformer-based architectures to analyze long-range dependencies among audio frames in the whole audio recording. Transformer-based models have proven to be useful for modeling dependencies among sequential data in a variety of tasks, related to both text [4, 16] and audio processing [1, 3]. They leverage a self-attention mechanism to model the relationships between tokens/frames, thereby building a global representation of the input sequence. Notice that acoustic events can occur at different temporal locations and have arbitrary durations. This makes their representations particularly well-suited for their identification of mosquitoes acoustic events. To our purposes, we generate audio signals containing acoustic events at different positions, which are then used to train the model for the prediction of frame-level labels. To the best of our knowledge, this is the first attempt to use transformer-based models to automatically identify mosquitoes acoustic events in audio recordings[1].

2 WAVEFORM ANALYSIS FOR MOSQUITO DETECTION

Detecting mosquito audio events in an audio recording entails the identification of the time intervals in which each event occurs. Hence, it entails modeling the audio sequence to take into account the sequential dependencies among the audio recordings.

We present *Wav*eform analysis for *Mos*quito *De*tection (WavMoDe). It extends the WavLM model [3] to address the task of mosquito event detection. WavLM [3] is a transformer-based model for modeling the sequential dependencies in an audio sequence. It has been originally proposed for speech-related tasks such as automatic speech recognition, speaker identification, and speaker

*All authors contributed equally to this research.

[1]The project source code is available for research purposes https://github.com/MorenoLaQuatra/ComParE2022_MED (Latest access: June 2022)

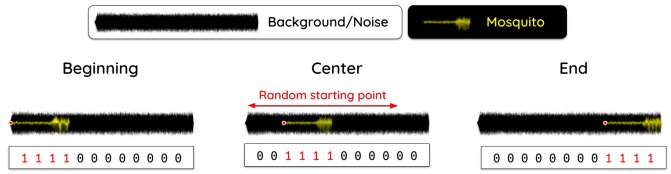

Figure 1: Frame-level annotation strategies for the training data.

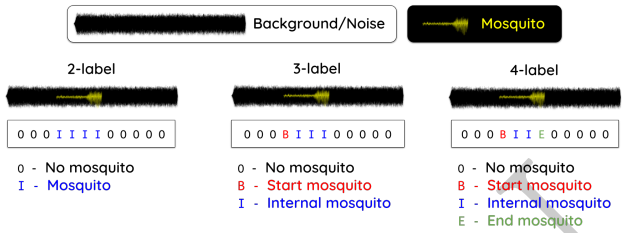

Figure 2: Mosquitoes masking procedures to train audio frame classification model.

diarization. It has shown to be effective for modeling the sequential dependencies of an audio sequence and we leverage the same architecture in this work to model mosquito sounds.

The original architecture is composed of a CNN-based feature extractor that transforms an input audio frame into a fixed-dimensional vector, followed by an encoder module composed of a stack of 12 transformer layers. Each audio frame has a fixed length of 20 ms with a sampling rate of 16kHz.

Similar to the original model, Wav-MoDe leverages the WavLM architecture for feature extraction and to model the sequential dependencies of an audio sequence. It is provided with an *audio frame classification* layer, on top of the original architecture, which is aimed at labelling each frame of an audio recording. The backbone architectures of Wav-MoDe and WavLM are specular. They are both pre-trained in a self-supervised manner on large-scale audio datasets.

2.1 Sequence tagging

The proposed methodology aims at tagging the input signal with a label indicating whether or not the corresponding frame contains an acoustic mosquito event. Given an audio recording, the goal is to identify the start and end timestamps of mosquito audio events. In order to train the sequence labeling model, a frame-level annotation of the signal is required.

Considering a set of audio recordings $X = \{x_1, x_2, \ldots, x_N\}$, each of their frames $f_j \in x_i$ is tagged with a label $y_j \in [0, L]$, where $y_j \neq 0$ if the frame contains an acoustic event, and $y_j = 0$ otherwise. $L \in \{1, 2, 3\}$ is the number of classes considered in the frame classification task (see Section 2.2 for a detailed explanation). When the frames are tagged at the frame level, the problem can be reformulated as a sequence labeling task in which each frame is assigned with a label. However, fine-grained annotations at the frame level are not only time-consuming but also challenging, since even for humans is difficult to identify frames that contain acoustic mosquito events. For this reason, prior works addresses mosquito audio event detection at a coarser granularity, i.e., at the recording level. In the latter scenario, y_j is a binary label indicating whether an event occurs in the audio recording.

To leverage the sequence-level context we generate artificial training signals including acoustic events positioned at different time points of the audio recording. In particular, we alter the original recording to insert acoustic events at specific recording time points. The signal is then labelled accordingly.

Figure 1 illustrates the methodology used for the generation of the training signals. Our approach is based on a data augmentation strategy that consists in inserting acoustic events at different positions in the recording. In particular, when we pick a non-mosquito recording from the training data, we do not apply any specific data augmentation and label each of its frames as $y_j = 0$. On the contrary, when we select a recording r_{mos}, that is labeled as *mosquito*

- we randomly select a non-mosquito recording r_{back} (e.g., an audio signal containing background noise),
- we select a random position p_{back} in the recording r_{back},
- we add r_{mos} to r_{back}, starting from the position p_{back}.

If the r_{back} recording is shorter than r_{mos}, then we produce a new sequence by repeating r_{back} until it is longer than r_{mos}. To generalize data augmentation is randomly applied with a probability of 75%, whereas in the remaining 25% cases we retain the original r_{mos} recording. In the former case, the position of the mosquito in the augmented recording is selected according to the following heuristics:

- in one third of the cases, the mosquito is added at the beginning of the recording (e.g., $p_{back} = 0$),
- in one third of the cases, the mosquito is added at the end of the recording (e.g., $p_{back} = len(r_{back}) - len(r_{mos})$),
- in one third of the cases, the mosquito is added at a random position; e.g., $p_{back} \sim \mathcal{U}(0, len(r_{back}) - len(r_{mos}))$,

Where $len(\cdot)$ denotes the length of the corresponding recording. The frames of the generated sequence are labeled as $y_j \neq 0$ for the ones that contain the mosquito and as $y_j = 0$ otherwise. More details about the labeling procedure for the generated recordings are given in Section 2.2. The background signals for the data augmentation process are randomly selected from the recordings of the training dataset containing background noise. Furthermore, additional noise samples are selected from

- FSDnoisy18k [7], a public-available dataset characterized by 42.5 hours of data belonging to 20 different classes. It contains real-world noisy audios and their corresponding manual annotations.
- Sound Events for Surveillance Applications (SESA) [15], a freely-accessible dataset containing 585 audios belonging to 4 different noise classes.
- Glasgow Isolated Sound Events (GISE-51) [17], an open dataset, based on FSD50K dataset [6], which includes 16'357 audios of different duration covering 51 different classes.

2.2 Data labeling

We combine audio signals containing mosquitoes events with background sounds to create new signals annotated with mosquito buzzing sounds. The audio frame classification task is treated as a sequence labeling problem. The label assigned to each audio frame indicates either the presence or the absence of a mosquito event. The proposed endeavor closely resembles the Named Entity Recognition (NER) task in the domain of Natural Language Processing: the key idea is to assign the label `O` to every frame that is not part of any event, and assign labels `B`, `I`, `E` to the frames at the beginning, inside, and at the end of the event, respectively.

In the experiments we test the following three different labeling options (see Figure 2):

- 2-label schema: It is the most straightforward labeling schema. The label `O` is assigned to every frame that is not part of any event whereas all the frames containing mosquito sounds are assigned to the class `I`.
- 3-label schema: The label `O` is assigned to every frame that is not part of any event. The frames containing mosquito sounds are assigned to the classes `B` if they are at the beginning of an event, `I` otherwise (i.e., if they are internal or at the end of the mosquito event).
- 4-label schema: It is similar to the 3-label schema, but the end of an event is explicitly highlighted using label `E`.

The 2-label schema is potentially an over-simplification and does not allow to distinguish between frames at the beginning, in the middle and at the end of the mosquito event. The 4-label schema, on the other hand, is a more fine-grained labeling schema and allows us to distinguish frames at the beginning, in the middle and at the end of the mosquito event. The 3-label scheme is a trade-off in which only the beginning of an event is denoted by the label `B` and the end is not explicitly indicated. During inference, we aggregate the predictions of all mosquito-related events, i.e., we sum the class probabilities of frames not labeled with `O` into one class probability (i.e, the overall probability of having a mosquito event). An empirical comparison between the labeling options can be found in Section 3.

3 EXPERIMENTS

We evaluate the Wav-MoDe performance on the official dataset (HumBugDB) of the ComParE 2022 mosquito sub-challenge [14]. However, notice that the proposed solution is general and can be applied to address similar tasks.

HumBugDB dataset characteristics. The dataset proposed for the mosquito sub-challenge is a large-scale collection of mosquito sounds recorded with mobile phones [9, 10]. It is characterized by $\sim$ 20 hours of recordings containing mosquito sounds and $\sim$ 15 hours of recordings of non-mosquito sounds. The dataset is split into train, dev-a, dev-b and test sets. While train and development sets are made available to the participants, the test set is redacted for a blind evaluation process. Both training and development sets contains only coarse-grained labels that indicate the presence of mosquito sounds in each audio sample.

Evaluation metrics. The ComParE 2022 mosquito sub-challenge relies on the PSDS [2] metric to compare the performance of the proposed systems. It is specifically designed to evaluate sound event detection systems. In light of the similarity between the mosquito event detection and the the speaker diarization problem, we also consider the Detection Error Rate (DER) [5], which is commonly used in the evaluation of binary classification tasks such as speech activity detection. It is defined as:

$$DER = \frac{F + M}{N} \tag{1}$$

where F, M, and N represent the duration of false positives, misses, and the total duration of each audio event, respectively.

Experimental settings. Wav-MoDe is trained on the training set given by the challenge organizers. The model is fine-tuned using AdamW optimizer [11] with an initial learning rate of 10^{-5} and a linear decay schedule with a decay rate of 0.01. We train the model for a maximum of 20 epochs using early stopping strategy. At each epoch, the model is evaluated on the development set provided by organizers (i.e., the evaluation is performed combining both the dev-a and dev-b sets) to choose the best checkpoint.

All the experiments were performed on a machine running Ubuntu 21.10 and equipped with AMD® Ryzen 9® 3950X CPU, Nvidia® RTX 3090 GPU, and 128 GB of RAM.

Attention window. Transformer-based models learn long-range dependencies among audio frames by encoding the entire sequence using an attention mechanism. However, the complexity of the attention mechanism is quadratic to the sequence length and impedes the model to process long audio recordings. To overcome the aforesaid limitation, Wav-MoDe is trained setting the *maximum* duration of the attention window to 60 seconds. During the inference process, the model is then applied to the recording by splitting it into chunks of different lengths $L_w \in \{60, 6, 0.6, 0.06\}$ seconds. The length of a single frame in Wav-MoDe is 0.02 seconds. Specifically, the highest value $L_w = 60$ seconds corresponds to the training time window, whereas $L_w = 0.06$ seconds corresponds to lowest resolution in terms of context that can be used to classify an audio frame.

3.1 Results

The results of the evaluations performed on the development and test sets are reported in Table 1. It indicates the DER and PSDS values achieved by Wav-MoDe in different evaluation settings, as well as for the baseline proposed in [8]. The results on the test set are limited by the maximum number of system submissions allowed by the challenge organizers.

We separately evaluate the results obtained by Wav-MoDe with different attention windows to assess the impact of contextual information on the results. For both metrics, the optimal results are obtained with $L_w = 0.6s$. This indicates that considering a larger context is not always beneficial. Indeed, the $L_w = 60s$ setting is likely to provide too much context for the classification task of a single frame, introducing noise and impeding the model to focus on the relevant context. On the other hand, by setting $L_w = 0.06s$ is likely to provide a too limited context for the classification task of a single frame, thus reducing the ability of the model to leverage the local context. The empirical evaluation shows that $L_w = 0.6s$ (i.e., the best performing configuration for both metrics

Table 1: Performance comparison of the baseline model and Wav-MoDe using different attention windows. Mel-BNN [8] represents the official baseline provided by task's organizers. *CI* indicates the bayesian confidence interval for the DER evaluation metric computed using a probability $\alpha = 0.9$. Symbols ↓ and ↑ imply lower and higher is better, respectively. The best results are highlighted in boldface.

Model	PSDS ↑		DER ↓		PSDS ↑	PSDS (MaxEFPR:3600) ↑
	Dev A	Dev B	Dev A	Dev B	Test	Test
Mel-BNN [8]	61.4	3.4	56.02 (CI: 54.50 - 57.54)	98.59 (CI: 97.47 - 99.71)	6.4	14.2
2-label-60ms	0.00	0.00	96.63 (CI: 96.49 - 96.78)	98.64 (CI: 98.36 - 98.93)	-	-
3-label-60ms	23.3	15.1	82.06 (CI: 81.45 - 82.67)	87.73 (CI: 85.89 - 89.57)	-	-
4-label-60ms	42.5	66.8	88.88 (CI: 88.50 - 89.26)	85.67 (CI: 83.53 - 87.81)	-	-
2-label-600ms	31.7	0.00	76.76 (CI: 75.83 - 77.69)	99.08 (CI: 98.26 - 99.90)	-	-
3-label-600ms	55.8	63.0	49.9 (CI: 48.2 - 51.6)	57.55 (CI: 51.35 - 63.74)	**48.9**	12.0
4-label-600ms	**68.7**	**81.0**	**48.0** (CI: 46.3 - 49.7)	**56.36** (CI: 50.00 - 62.72)	24.6	**36.3**
2-label-6s	5.6	0.00	92.3 (CI: 91.3 - 93.1)	100	-	-
3-label-6s	11.6	49.4	50.0 (CI: 48.2 - 51.8)	60.3 (CI: 54.2 - 66.5)	1.2	12.0
4-label-6s	8.1	17.0	56.7 (CI: 55.1 - 58.3)	82.2 (CI: 77.2 - 87.2)	-	-
2-label-60s	5.5	0.00	92.0 (CI: 91.1 - 93.0)	99.7 (CI: 99.2 - 1.00)	-	-
3-label-60s	21.1	36.9	49.8 (CI: 48.1 - 51.6)	65.6 (CI: 59.4 - 71.9)	0.3	0.5
4-label-60s	12.3	4.3	55.8 (CI: 54.2 - 57.4)	82.4 (CI: 76.9 - 87.8)	-	-

in the development sets) is a good trade-off, as provides enough context without introducing significant irrelevant information.

Among different labeling schemes, the best results are obtained with the 3-/4-label scheme. By explicitly indicating the beginning and ending of an event, the model is better trained to recognize local patterns. Comparing their results in the development sets, 3-label scheme provide better results with larger attention windows, while 4-label scheme provide better results with smaller ones.

Pitfall in evaluation metrics. The official evaluation metric (PSDS) provides a quantitative assessment of the ability of the model to recognize mosquito events. However, when evaluating the model from a qualitative perspective, we found that it is likely to predict short-lasting background within long-lasting mosquito events. The PSDS metric, in its standard setting overlooks this behaviour and, as a result, both model fine-tuning and evaluation are likely to lead to sub-optimal results in real-world applications. To overcome the aforesaid limitation, contest organizers provided an alternative PSDS scoring function, which accounts for short incorrect detections (see the PSDS column with MaxEFPR=3600 in Table 1). These modifications penalizes Wav-MoDe, whose results, due to the model's tendency to produce short-lasting background detections, are significantly lower compared to the original scoring system (see the comparison between PSDS settings in Table 1).

The results on the test set shows that 3-label scheme with $L_w = 0.6$ seconds achieves the best performance considering the standard PSDS setting, while 4-label scheme with $L_w = 0.6$ seconds performs best considering modified PSDS setting.

4 CONCLUSIONS AND FUTURE WORKS

In this work, we proposed a novel approach, namely Wav-MoDe, to detect and localize mosquito events from audio recordings. It is based on an end-to-end transformer architecture that is trained to predict the mosquito event label for each frame of the input waveform. Although the model achieves very good performance according to the standard evaluation metrics, unfortunately, it is susceptible to predicting short-lasting background within long-lasting mosquito events. This behaviour is partly highlighted by the alternate PSDS scoring provided by the contest organizers, which penalizes systems providing such prediction pattern.

The future research activities call for the study and adoption of new and ad hoc evaluation metrics able to effectively manage short-lasting events and thus better reflecting human expectations. Furthermore, we aim at overcoming the limitations enforced by the complexity of the standard attention mechanism by leveraging time-restricted self-attention models [12].

Finally, seeking for a more efficient utilization of temporal correlations, we will investigate the design of streaming architectures able to provide online prediction of mosquito events.

ACKNOWLEDGMENTS

The research leading to these results has been partly funded by the SmartData@PoliTO center for Big Data and Machine Learning technologies.

REFERENCES

[1] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems* 33 (2020), 12449–12460.

[2] Çağdaş Bilen, Giacomo Ferroni, Francesco Tuveri, Juan Azcarreta, and Sacha Krstulović. 2020. A framework for the robust evaluation of sound event detection. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 61–65.

[3] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. 2021. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *arXiv preprint arXiv:2110.13900* (2021).

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and*

Short Papers). Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[5] Jonathan G Fiscus, Jerome Ajot, Martial Michel, and John S Garofolo. 2006. The rich transcription 2006 spring meeting recognition evaluation. In *International Workshop on Machine Learning for Multimodal Interaction*. Springer, 309–322.

[6] Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra. 2021. Fsd50k: an open dataset of human-labeled sound events. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 30 (2021), 829–852.

[7] Eduardo Fonseca, Manoj Plakal, Daniel PW Ellis, Frederic Font, Xavier Favory, and Xavier Serra. 2019. Learning sound event classifiers from web audio with noisy labels. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 21–25.

[8] Ivan Kiskin, Adam D Cobb, Marianne Sinka, Kathy Willis, and Stephen J Roberts. 2021. Automatic Acoustic Mosquito Tagging with Bayesian Neural Networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 351–366.

[9] Ivan Kiskin, Marianne Sinka, Adam D Cobb, Waqas Rafique, Lawrence Wang, Davide Zilli, Benjamin Gutteridge, Rinita Dam, Theodoros Marinos, Yunpeng Li, et al. 2021. HumBugDB: A Large-scale Acoustic Mosquito Dataset. *arXiv preprint arXiv:2110.07607* (2021).

[10] Ivan Kiskin, Lawrence Wang, Marianne Sinka, Adam D. Cobb, Benjamin Gutteridge, Davide Zilli, Waqas Rafique, Rinita Dam, Theodoros Marinos, Yunpeng Li, Gerard Killeen, Dickson Msaky, Emmanuel Kaindoa, Kathy Willis, and Steve J. Roberts. 2021. *HumBugDB: a large-scale acoustic mosquito dataset*. https://doi.org/10.5281/zenodo.4904800 Funding from the 2014 Google Impact Challenge Award, and The Bill and Melinda Gates Foundation (https://www.gatesfoundation.org/about/committed-grants/2019/07/opp1209888).

[11] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Bkg6RiCqY7

[12] Daniel Povey, Hossein Hadian, Pegah Ghahremani, Ke Li, and Sanjeev Khudanpur. 2018. A time-restricted self-attention layer for ASR. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5874–5878.

[13] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. 2014. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*. 1041–1044.

[14] Björn W. Schuller, Anton Batliner, Shahin Amiriparian, Christian Bergler, Maurice Gerczuk, Natalie Holz, Pauline Larrouy-Maestri, Sebastian P. Bayerl, Korbinian Riedhammer, Adria Mallol-Ragolta, Maria Pateraki, Harry Coppock, Ivan Kiskin, Marianne Sinka, and Stephen Roberts. 2022. The ACM Multimedia 2022 Computational Paralinguistics Challenge: Vocalisations, Stuttering, Activity, & Mosquitos. In *Proceedings ACM Multimedia 2022*. ISCA, Lisbon, Portugal. to appear.

[15] Tito Spadini. 2019. *Sound Events for Surveillance Applications*. https://doi.org/10.5281/zenodo.3519845

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[17] Sarthak Yadav and Mary Ellen Foster. 2021. GISE-51: A scalable isolated sound events dataset. https://doi.org/10.48550/ARXIV.2103.12306