Boes and Van hamme
EURASIP Journal on Audio, Speech, and Music Processing (2022) 2022:25
https://doi.org/10.1186/s13636-022-00252-9

EURASIP Journal on Audio, Speech, and Music Processing

METHODOLOGY Open Access

Multi-encoder attention-based architectures for sound recognition with partial visual assistance

Wim Boes* and Hugo Van hamme

Abstract

Large-scale sound recognition data sets typically consist of acoustic recordings obtained from multimedia libraries. As a consequence, modalities other than audio can often be exploited to improve the outputs of models designed for associated tasks. Frequently, however, not all contents are available for all samples of such a collection: For example, the original material may have been removed from the source platform at some point, and therefore, non-auditory features can no longer be acquired. We demonstrate that a multi-encoder framework can be employed to deal with this issue by applying this method to attention-based deep learning systems, which are currently part of the state of the art in the domain of sound recognition. More specifically, we show that the proposed model extension can successfully be utilized to incorporate partially available visual information into the operational procedures of such networks, which normally only use auditory features during training and inference. Experimentally, we verify that the considered approach leads to improved predictions in a number of evaluation scenarios pertaining to audio tagging and sound event detection. Additionally, we scrutinize some properties and limitations of the presented technique.

Keywords: Multi-encoder, Transformer, Conformer, Sound recognition, Audio tagging, Sound event detection, Multimodal data, Audiovisual data, Missing data

1 Introduction

Numerous sounds carry meaning relevant to everyday life: Speech is a particularly important subclass, but more general acoustic events, such as the screaming of a baby or the ringing of an alarm bell, can obviously also be of great importance to humans. In this light, it is only logical that sound recognition tasks are quickly becoming significant machine learning subjects.

The most prominent related topics are aggregated in a yearly contest, the Detection and Classification of Acoustic Scenes and Events (DCASE) challenge. Its most recent version [1] featured multiple subtasks revolving around classification of environmental events: If coupled with estimation of temporal boundaries, this problem is usually called sound event detection, else, it is typically referred to as audio tagging. Other subjects include but are not limited to spatial localization and automated captioning of auditory inputs.

The research presented in this project deals with the integration of visual information into sound recognition systems. Prior efforts have shown that this process can lead to enhanced predictions: [2–4] have shown that employing audiovisual variants of deep learning architectures, such as feedforward and attention-based neural networks, can be beneficial for audio tagging. In [5], it is shown that applying early (feature) fusion to convolutional recurrent models can provide improvements for sound event detection as well.

Many commonly used data sets for sound recognition tasks in some way stem from Audio Set [6], which is a very large-scale cluster of auditory segments originated from YouTube. These clips are weakly labeled in the sense

*Correspondence: wim.boes@esat.kuleuven.be
ESAT, KU Leuven, Leuven, Belgium

that the semantic categories of occurring audio events are given, but other details, such as their on- and offsets, are excluded. Derivations usually retain only a manually controlled subset of the full collection and occasionally append extra information: As an example, AudioCaps [7] attaches multiple descriptive text captions to each comprised sample.

As a consequence of much sound recognition data having its origin in YouTube, a multimedia library, it is straightforward to retrieve videos and eventually incorporate these into systems designed for related tasks. This is the approach taken by works mentioned earlier in this section. However, this procedure runs into one specific issue: Content may have been removed from this platform at some point in time, e.g., because of account deletions or copyright claims. Consequentially, the source material can not necessarily be utilized to perform computations for all samples of the considered collection. On the acoustic side, this is usually not a problem: Curators of data sets derived from Audio Set [6] often ensure availability of all audio snippets by maintaining a separate database of copies. Having said that, for the visual information, this kind of careful conservation is unfortunately not customary.

This situation can be regarded as an intense expression of the problem of missing data. As outlined in [8], one possible approach to this matter is to simply discard all data entries with absent values. This is also the route that has been taken in the previously referenced projects related to sound recognition with visual assistance. Another, less destructive option involves imputation or replacement of missing entries. This technique can be utilized in conjunction with deep learning models, such as generative adversarial networks [9], and currently occupies most of the literary space. Applications can be found in, among others, the domains of healthcare [10], biomedical data mining [11], recommendation systems [12] and speech recognition [13].

The downside to imputation techniques is that, in order for them to work properly, they require quite strong assumptions about the statistics of the complete data [14]. For example, many algorithms are based on the so-called missing at random condition: In this hypothesis, the absent values are connected to the known entries through variables which may or may not be hidden. This conjecture can undoubtedly be justified in some cases, such as well-structured time series, but is definitely not universally applicable.

In the context of this work, these assumptions are hard to defend: We deal with noisy audiovisual clips for which the two modalities are relatively decoupled. There are two frequently occurring issues in this regard. Firstly, while the sound is guaranteed to be accurately labeled, the curators of the used data set make no such promises for the visual component. For instance, the videos of some samples consist of nothing more than a meaningless still image or even a black screen. Secondly, even for the examples with real visual information, there is often a severe lack of semantic or temporal synchronization between the two streams.

Nevertheless, in this project, we still seek to tackle this specific manifestation of the problem of missing values related to sound recognition based on audiovisual clips originated from YouTube. To this end, we stray away from the unsatisfactory deletion procedure, which has been employed in previous works on the considered subject, and imputation, which is unrealistic in this instance as explained before. Instead, we propose an approach based on dynamically weighted fusion of intermediary auditory and visual features. This is done by adapting the multi-encoder framework presented in [15], which was originally utilized to achieve more robust predictions for speech recognition, without necessarily increasing time and/or memory complexity.

This work is organized as follows: In Section 2, we elaborate upon the proposed method. In Section 3, we go into the performed experiments: We provide a detailed description of the used setup and analyze obtained results. Finally, in Section 4, we summarize the most important conclusions of the conducted research.

2 Method

In this section, the proposed method is discussed. We adapt and extend the middle fusion approach presented in [15] to produce a process that allows us to combine auditory inputs with partially available visual features, during training as well as inference. Section 2.1 goes into the attention-based neural networks which are employed as base components of the considered systems. Next, in Section 2.2, the multimodal multi-encoder learning framework is fully explained.

2.1 Attention-based architectures

In this section, we elaborate upon the attention-based neural networks which are used throughout this work. First, we provide a detailed description of the transformer [16], which at first was used to tackle machine translation but has since been used for many other purposes. Afterwards, we examine the conformer model [17], which augments the previously mentioned architecture with convolutional blocks and was originally designed for automatic speech recognition. In the outline below, these systems are presented in concrete forms suitable for the tasks relevant to this work, i.e., audio tagging and sound event detection.

2.1.1 Transformer model

The transformer for joint audio tagging and sound event detection is schematically illustrated in Fig. 1. It strongly resembles the system described in [18], which itself is a variant of the original model [16].

The transformer is a neural network that converts sequential inputs into a series of probabilities. In the context of this project, these values indicate which sounds are present during each time frame of an audio(visual) recording. Similar to the approach taken in the BERT model [19], built for natural language processing tasks, we append a learnable classification token to the set of features at the decoder side. Because of this change, the output of the system will contain an extra vector, which can be used to obtain clip-level predictions.

The encoder and decoder blocks of the transformer model (in this project, we use 3 of each) closely resemble each other. Their architectures consist of a combination of layer normalization operations [20], residual connections [21], feedforward components — which, in this case, are made up of two consecutive layers with 512 ReLU and 128 linear neurons respectively — and perhaps most importantly, multi-head attention modules. Also, dropout [22] with a rate of 0.1 is used after each of the aforementioned feedforward submaps, this is not explicitly shown in Fig. 1.

The multi-head attention mechanism [16] performs a content-based comparison between two sets of features, referred to as queries and keys respectively. This is done by computing scaled dot products. Afterwards, the resulting so-called attention weights are multiplied with another series of inputs, namely, the values. This operation is mathematically expressed in Eq. 1.

$$A(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

In Eq. 1, Q, K and V are the matrices containing queries, keys and values respectively. The scaling parameter in the denominator of the softmax function, d_k, refers to the size of the keys. In this project, we ensure this number is equal to 128 at all times.

The multi-head module extends the simple attention calculation given in Eq. 1 by applying linear mappings to the queries, keys and values before this computation and repeating the entire process several times. Eventually, the outcomes of the so-called multiple (in this work, we stick to 3) attention heads are concatenated and fed through a last projection layer with 128 units to obtain the final result. A practical detail to be added to this description is that dropout [22] with a rate of 0.1 is also applied to all attention weights and outputs of the considered neural network blocks.

All components of the transformer model that have been discussed this far perform content-based computations and ignore sequential information. This shortcoming is mitigated by adding learnable positional encodings [23] to the inputs: These are trainable vectors representing the absolute (temporal) locations of all frames in the used feature sequences.

Previously, transformer encoders have successfully been used to tackle sound event detection [24]. When only one set of inputs is employed, as is the case for the cited work, it does indeed not make sense to utilize the full structure: The addition of a decoder would not add functionality but only increase model complexity. However, in this project, we attempt to exploit multiple sequences of features, and in that case, using the complete transformer as a basis is more appropriate. This is further elaborated upon in Section 2.2.

2.1.2 Conformer model

The conformer encoder [17] is an extension of the corresponding component in the transformer. Compared to

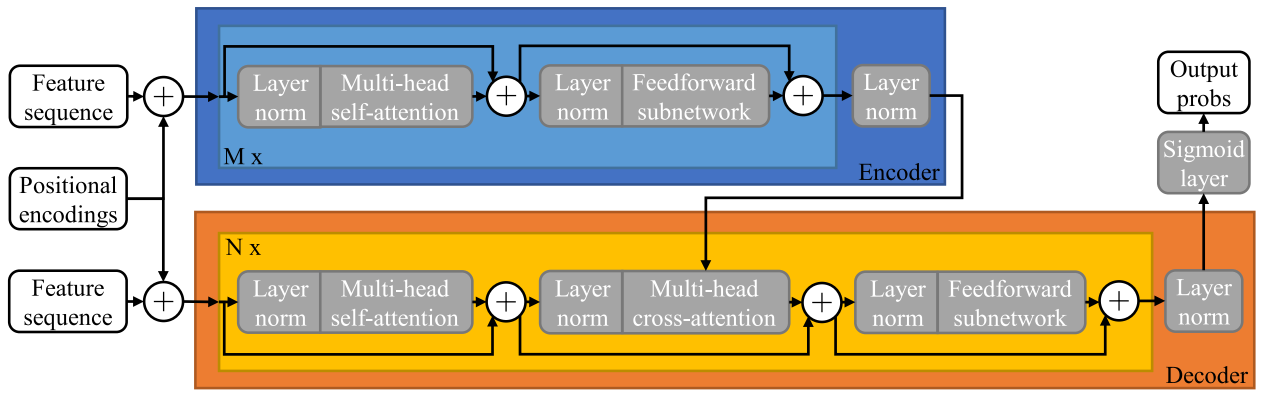

Fig. 1 Diagram of transformer model for sound recognition

the latter architecture, which focuses on finding global dependencies between data frames, modules are added to substantially improve its ability to perform local processing. Figure 2 depicts an appropriately designed model [24] based on this deep learning entity for joint audio tagging and sound event detection.

A full explanation of all components in this model can be found in the original work [17]. In what follows, we mainly focus on the surface-level differences between transformer and conformer encoders, while ignoring some less important and finer details.

Firstly, the feedforward submodule in the transformer encoder is replaced by two such structures in the conformer variant. The associated residual connections are foreseen of a halving operation, akin to the Macaron neural network proposed in [25]. Also, instead of a rectified linear unit (ReLU), the more complex Swish activation function [26] is employed.

Secondly, the multi-head attention block explained in the previous section is exchanged for a version that also incorporates relative positional encodings [27]: By injecting these (learnable) embeddings, representing relative distances between feature vectors, the ability of this module to perform location-based (as opposed to content-based) processing is notably augmented.

Thirdly, and perhaps most importantly, a convolutional subnetwork is inserted into the architecture. This component is what allows the conformer encoder to perform local computations, in contrast to the transformer variant, which is only capable of dealing with global (and mostly content-based) dependencies. This extra module consists of the following sequence of operations: pointwise convolution coupled with a gated linear unit [28], depthwise convolution (with a kernel size of 7, as in [24]), batch normalization [29] with a momentum of 0.9, application of the Swish activation function [26], another pointwise convolution and finally, dropout [22] with a rate of 0.1.

Conformer encoders were originally designed for automatic speech recognition but, as demonstrated, have also been employed for sound recognition [24]. This neural structure cannot deal with more than one series of inputs, which is necessary for our purposes. Luckily, it can easily be expanded into a more appropriate encoder-decoder architecture by following the blueprint of the transformer model shown in Fig. 1.

In the rest of this work, when we mention the conformer system, we refer to the model of which a simplified diagram is drawn in Fig. 3. It is structurally similar to the architecture in Fig. 1, but instead of transformer-based components, it utilizes a conformer encoder and decoder. The latter building block is derived from the former by simply adding a multi-head relative cross-attention module (with corresponding layer normalization and residual connection) after the self-attention variant, analogously to the transformer.

2.2 Multimodal multi-encoder learning framework

In this section, we discuss the proposed multimodal multi-encoder learning framework. To this end, we first review the original middle fusion algorithm described in [15]. Afterwards, we explain the adjustments that have to be made to ensure the resulting approach is suited to tackle the problem at hand, i.e., sound recognition with partial visual assistance.

Figure 4 shows a simplified diagram of the middle fusion multi-encoder learning framework. Compared to the base architectures discussed in Section 2.1, the cross-attention modules of all decoder blocks are duplicated a number of times, depending on the amount of input sequences supplied to corresponding encoder structures. The vectors produced by these copies are interpolated in a linear fashion to obtain intermediate features which are used downstream in the model.

In [15], this principle is applied in the context of automatic speech recognition: Features representing

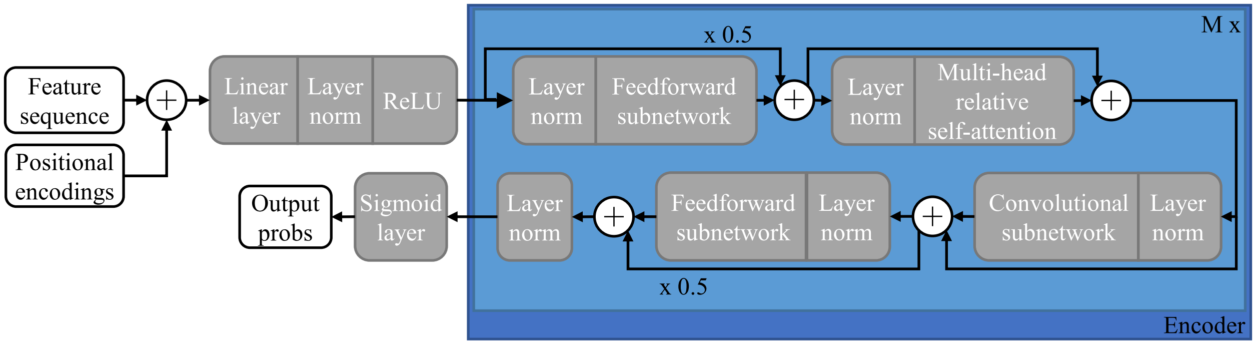

Fig. 2 Diagram of conformer encoder model for sound recognition

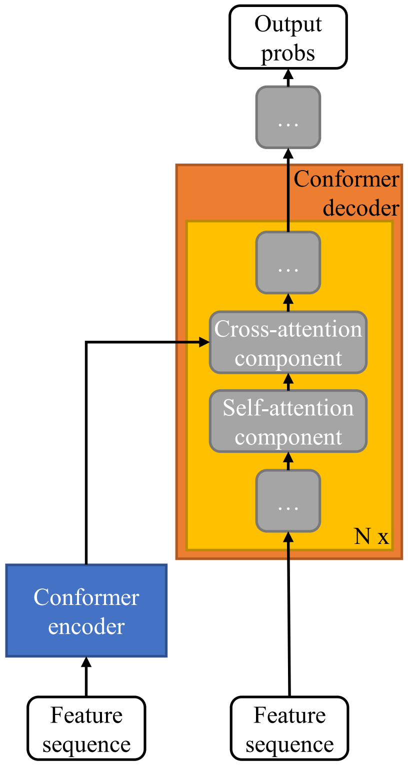

Fig. 3 Simplified diagram of encoder-decoder conformer model for sound recognition

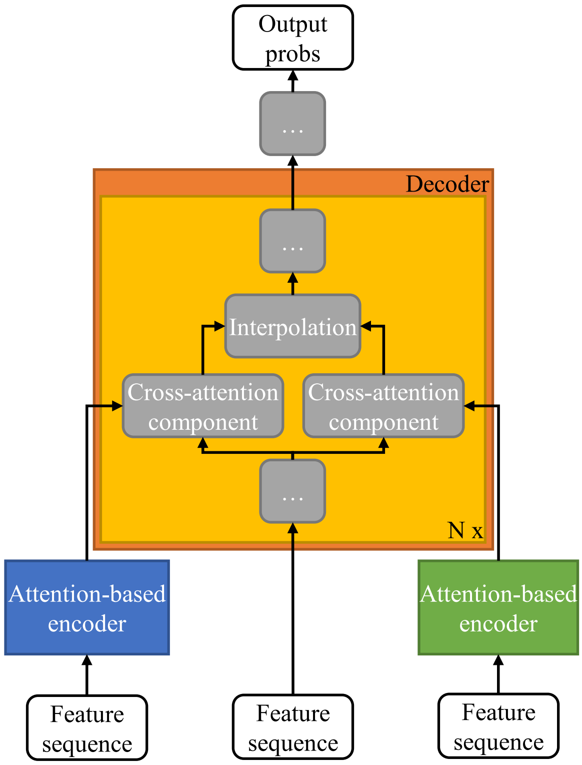

Fig. 4 Simplified diagram of multi-encoder framework

the spectral magnitude and phase components of the used auditory data are combined using this multi-encoder approach to enhance the performance of the decoder, which is charged with the task of transforming input characters into output token probabilities. Fixed interpolation weights are used for calculating the relevant convex sums, biased towards the (generally) more salient magnitude representations. The cited work also investigates other configurations, such as a late fusion version and a variant with tied encoder parameters, which can be used to limit memory complexity. Preliminary experiments have shown that these options do not provide additional benefits in the context of this project, and thus, they are not discussed further.

As explained in Section 1, we want to build a system that can perform sound recognition using multimodal data, of which the auditory component is always at hand, but the visual information is only partially available. To this end, we adopt the framework depicted in Fig. 4, but in contrast to the original approach, we set the multi-encoder interpolation parameters dynamically, both during training and testing: When series of features, i.e., those related to vision, are missing, the weights associated with these sequences are set to 0. Naturally, the others are scaled in order for the total to remain equal to 1. In Section 3, we also explore a novel weighting scheme for the learning phase.

3 Experiments

We detail the performed experiments involving the proposed multi-encoder framework, elaborated upon in Section 2, meant to deal with the considered problem of missing values. In Section 3.1, the setup is discussed, while Section 3.2 analyzes the obtained results.

3.1 Setup

In this section, we fully lay out the experimental setup. Firstly, the used data set and the features that are extracted from this collection are elaborated upon. Next, we report the postprocessing steps applied to convert the output probabilities of the models described in Section 2 into binary predictions, and we go into the metrics employed to gauge the performance of the examined systems. Finally, for completeness, we list all relevant training and testing hyperparameters.

3.1.1 Data

In this work, we use the data associated with task 4 [30] of the DCASE 2020 challenge [31]. This large-scale set contains recordings with a maximum length of 10 seconds. Each instance features a number of potentially overlapping auditory events belonging to the 10 possible environmental classes itemized in Table 1. The collection is split into multiple partitions. The amount of samples in each subset is listed in Table 2.

Details about the occurring auditory events are provided in different ways for the distinct collections mentioned in Table 2. The strongly supervised training, validation and evaluation partitions encompass full annotations of all occurring sounds, including on- and offsets. The samples of the other subsets are either weakly supervised, i.e., only clip-level labels are available, or simply do not contain any relevant information at all.

The recordings in the strongly supervised training set originate from Freesound, a collaborative database of sounds, and only contain auditory information. All other clips come from YouTube, a very well-known large-scale multimedia library. For most of these examples, visual information can also be extracted. However, as thoroughly explained in Section 1, this is not possible for all data points: On average, for 15.5% of those samples, the video can no longer be downloaded.

3.1.2 Features

In this project, we investigate multiple ways of preprocessing the auditory and visual streams of the considered data. In Section 3.2, the discussion on the results of the experiments includes an analysis of which features lead to improvements under the proposed framework for a number of evaluation scenarios.

Spectral auditory features We first resample the audio streams to 16 kHz and apply peak amplitude normalization. Next, log mel magnitude spectrograms with 64 frequency bins are extracted using a window size of 1024 samples (corresponding to 64 ms) and a hop length of 313 samples (corresponding to 20 ms). For a clip of 10 seconds, this results in 512 frames. Lastly, per frequency bin standardization is performed, based on the statistics of the training data.

Table 1 Labeled sound categories in data set

Speech	Frying
Dog	Blender
Cat	Running water
Alarm/bell/ringing	Vacuum cleaner
Dishes	Electric shaver/toothbrush

Table 2 Number of samples per partition in data set

Data partition	Number of recordings
Unsupervised training	14412
Weakly supervised training	1578
Strongly supervised training	2584
Validation	1168
Evaluation (public)	692

Compared to the other (pretrained) embeddings described below, these features are fairly rudimentary. To create higher-level auditory vectors which can be supplied as inputs to the models described in Section 2, we use a feature extractor with a similar architecture as in the baseline model [32] for task 4 [30] of the DCASE 2020 challenge [31]. It is a convolutional network which is made up of seven consecutive stacks, which each perform the following five operations: convolution, batch normalization [29] with a momentum of 0.9, application of the ReLU function, dropout [22] with a rate of 0.1 and average pooling. The hyperparameters of the convolutional layers are listed per block in Table 3. In Table 3, the first and second numbers of each tuple in the columns on kernel size and stride relate to time and frequency axes respectively. As can be inferred from this list, this convolutional feature extractor reduces the frequency dimension of the spectral map to one and has a total temporal pooling factor of 8: For a clip of 10 seconds, this results in the original series being transformed into a sequence of 64 vectors. This convolutional feature extractor is inserted into the models described in Section 2 and the complete architecture is trained in an end-to-end manner.

OpenL3 visual features OpenL3 [33] is an embedding model designed to predict correspondence between auditory and visual streams, trained in a self-supervised way. It is pretrained on Audio Set [6].

To obtain pretrained visual features, still images are first sampled from the available visual streams at a rate of about 6.5 fps. The frames are fed into the video subnetwork of OpenL3 [33]. For a recording of 10 seconds, these steps lead to 64 512-dimensional vectors.

Temporally coherent visual features Still images are first sampled from the visual streams at a rate of about 6.5 fps. Next, these frames are passed through the video embedding model described in [34], trained in a self-supervised manner using a loss optimizing temporal coherency. For a clip of 10 seconds, this procedure results in a series of 64 2048-dimensional vectors.

Table 3 Hyperparameters of convolutional feature extractor

Block	Operation	Channels	Kernel size	Strides
0	Convolution	16	(3, 3)	(1, 1)
	Pooling	-	(2, 2)	(2, 2)
1	Convolution	32	(3, 3)	(1, 1)
	Pooling	-	(2, 2)	(2, 2)
2	Convolution	64	(3, 3)	(1, 1)
	Pooling	-	(2, 2)	(2, 2)
3-4-5-6	Convolution	128	(3, 3)	(1, 1)
	Pooling	-	(1, 2)	(1, 2)

VGG16 visual features Still images are sampled from the visual streams at a rate of about 6.5 fps. The resulting frames are fed into VGG16 [35], a convolutional model for image classification, pretrained on the ImageNet data set [36]. The 4096-dimensional outputs of the last feed-forward layer of this neural network are used as feature sequences in this project. For a recording of 10 seconds, these steps lead to 64 vectors.

3.1.3 Postprocessing

To calculate clip-level metrics, the relevant probabilities are converted into binary decisions by employing class-wise thresholds, optimized on the validation partition of the employed data set. The search space for these hyperparameters is restricted to the linear span ranging from 0.1 to 0.9 in steps of 0.1.

To obtain sound event detection scores, the frame-level probabilities are transformed via the following process: Firstly, they are binarized, and secondly, the decisions are passed through a median smoothing operation. The hyperparameters associated with these steps are (separately) optimized on the validation partition of the employed data set, per sound category as well as per used evaluation metric. The search space for the thresholds is limited to a linear span ranging from 0.1 to 0.9 in steps of 0.1. For the filter sizes, values from 1 to 31 are tested in increments of 2.

Other postprocessing operations can be applied to further enhance the predictions of the considered systems. For example, in [24], model ensembling is employed in the form of score-level fusion. In this work, we choose not to take any further steps as the number of possibilities is seemingly unlimited. Also, the main goal of this project is not necessarily to achieve the perfectly optimized model, rather, we want to demonstrate the effectiveness of the proposed method.

3.1.4 Metrics

To evaluate the considered models for both audio tagging and sound event detection, we use a variety of metrics representing distinct scenarios, involving different requirements in terms of temporal localization. These scores are specifically chosen because of their prevalence in the field of sound recognition.

Clip-based F1 score (CBF1) For audio tagging, we use the micro-averaged clip-based F1 score [37]. This metric was also used in the DCASE 2017 challenge [38].

Segment-based F1 score (SBF1) We use the micro-averaged segment-based F1 score based on slices of 1 s [37] to quantify the effectiveness of the predictions of the proposed models. Because of the relatively long slice length, this metric is suitable for investigating coarse-grained sound event detection performance. It was also utilized in the DCASE 2017 challenge [38].

Event-based F1 score (EBF1) We use the macro-averaged event-based F1 score with tolerances of 200 ms for onsets and 20% of the lengths of the audio events (up to a maximum of 200 ms) for offsets [37]. Because of the strict localization requirements, this metric is suitable for investigating fine-grained sound event detection performance. It was also utilized in the DCASE 2018 [39], 2019 [40] and 2020 [31] challenges.

Polyphonic sound event detection scores (PSDS) We use two polyphonic sound event detection scores [41], representing distinct evaluation scenarios. In the rest of this work, they are referred to as PSDS1 and PSDS2. The former imposes strict requirements on the temporal localization accuracy, the latter is more lenient in this regard. The hyperparameters for these measures are summarized in Table 4. One of the default postprocessing steps described before is left out in this case: These scores are computed using 50 fixed operating points, in which thresholds linearly distributed from 0.01 to 0.99 (with a step size of 0.02) are used to convert probabilities into binary decisions. These metrics were also utilized in the DCASE 2021 [1] challenge.

3.1.5 Hyperparameters

In this section, we provide details on the used training and testing procedures for the sake of completeness. PyTorch [42] is utilized to implement all of the work.

Elaborate explanations of the selection processes for the encoder interpolation weights of the considered models are omitted from this section, as these procedures vary per experimental setting. Instead, these descriptions are left for the relevant parts in Section 3.2.

Training As already outlined in Section 3.1, the data employed in this research project is heterogeneously

Table 4 PSDS hyperparameters

Hyperparameter	PSDS1	PSDS2
Detection tolerance criterion	0.7	0.1
Ground truth intersection criterion	0.7	0.1
Cross-trigger tolerance criterion	N/A	0.3
Cost of class instability	1	1
Cost of cross-triggers	0	0.5
Maximum false positive rate	100	100

annotated, and thus, combining all available samples into the learning procedure is challenging, especially when it comes to the unlabeled instances. To deal with this difficulty, mean teacher training [43] is performed. In this framework, two models called the student and the teacher are utilized. They share the same architecture, but their parameters are updated differently.

The student system is trained regularly, i.e., a differentiable objective is minimized. However, the weights of the teacher are computed as the exponential moving average of the student parameters with a multiplicative decay factor of 0.999 per training iteration. The loss employed to train the student consists of four terms: The first two are clip-level and frame-level binary cross entropy functions, which are only computed for the weakly and strongly labeled clips respectively. The other components are mean-squared error consistency costs between the clip-level and frame-level output probabilities of the student and teacher models, which can be computed for all data, including the unannotated samples. The classification and consistency terms are summed with weights 1 and 2 respectively to obtain the final objective.

During training, data augmentation is also employed in the form of mixup [44], which comes down to creating extra learning examples (and associated labels) by linearly interpolating the original samples. We use this method with an application rate of 33%. The mixing ratios are randomly sampled from a beta distribution with shape parameters equal to 0.2.

Models are trained for 100 epochs. Per epoch, 250 batches of 128 samples are given to the networks. Each batch contains 32 strongly labeled, 32 weakly labeled and 64 unlabeled examples. Rectified Adam [45, 46] is employed to train the weights of the student systems. Learning rates start at 0.001 and decay multiplicatively with a factor of 0.1 per 10000 iterations.

Testing Metrics are calculated on probabilities produced by student models after the last training epoch.

3.2 Results

In this section, all experimental results are listed and analyzed. We report evaluation scores which have been averaged over 20 training runs (with independent initializations of all model parameters) to ensure reliability, as well as the associated standard deviations.

Preliminary experiments have indicated that architectures only employing visual features underperform badly with regard to sound event detection. These outcomes are in line with findings divulged in prior research [5]. Clearly, on their own, these pretrained vectors are not able to properly perform temporal segmentation on the considered multimodal data. As a consequence of this observation, the following design choice has been made: All of the explored systems take spectral auditory maps as inputs to their decoders and to one of their encoders. In other words, in what follows, models without acoustic features are not considered.

3.2.1 *Uni-encoder attention-based models*

Table 5 contains the results obtained by baseline models which do not utilize any visual information at all and thus do not run into the examined missing data problem: Each of these systems uses auditory features as inputs to its decoder as well as its single encoder.

In [24], transformer and conformer encoders have been used in a very similar way to tackle sound event detection on the same data set. The performance values reported in the cited work for such models that do not use any type of ensemble method are comparable to those in Table 5. The remaining small disparities can partially be attributed to architectural differences, since the referenced systems do not include decoder components, unlike is the case in this project.

Interestingly, models using the transformer architecture as a base outperform those using conformer components in terms of most considered performance metrics, in contrast to what is reported in [24]. This trend will reappear in the results of systems exploiting visual information (or equivalently, using multiple encoders), which are discussed in the following sections. However, the differences are too small and/or inconsistent to infer conclusions on the superiority of one or the other.

3.2.2 *Bi-encoder attention-based models*

Table 5 also lists the scores obtained by models that use two encoders: one takes in spectral auditory maps, the other accepts a type of pretrained visual features.

The interpolation weights associated with these two encoders are determined by maximizing the performance

Table 5 Results of uni- and bi-encoder attention-based models

Base model	Encoder inputs	Encoder weights		CBF1	SBF1	EBF1	PSDS1	PSDS2
		Training	Inference	(%)	(%)	(%)		
Transformer	Spectral auditory features	1	1	82.48	78.92	49.43	0.4083	0.6413
				±0.47	±0.48	±0.61	±0.0080	±0.013
	Spectral auditory features	0.5	0.75	**84.57**	**80.56**	**51.00**	0.4168	**0.6567**
	OpenL3 visual features	0.5	0.25	±0.40	±0.40	±0.55	±0.0080	±0.011
	Spectral auditory features	0.75	0.875	84.10	80.11	50.97	**0.4188**	0.6502
	Temporally coherent visual features	0.25	0.125	±0.45	±0.46	±0.58	±0.0086	±0.014
	Spectral auditory features	0.5	0.875	84.22	79.91	48.28	0.3891	0.6144
	VGG16 features	0.5	0.125	±0.41	±0.41	±0.60	±0.0079	±0.011
Conformer	Spectral auditory features	1	1	82.52	78.62	48.41	0.3849	0.6451
				±0.40	±0.41	±0.57	±0.0086	±0.011
	Spectral auditory features	0.75	0.875	83.83	80.22	**50.41**	**0.4010**	**0.6501**
	OpenL3 visual features	0.25	0.125	±0.44	±0.45	±0.64	±0.0090	±0.012
	Spectral auditory features	0.75	0.875	**84.15**	**80.48**	50.18	0.4000	0.6490
	Temporally coherent visual features	0.25	0.125	±0.42	±0.38	±0.63	±0.0092	±0.011
	Spectral auditory features	0.5	0.875	83.83	79.43	46.48	0.3615	0.6226
	VGG16 features	0.5	0.125	±0.43	±0.48	±0.60	±0.0079	±0.014

of the models on the validation partition of the data set at hand. Specifically, we choose the hyperparameters which lead to the highest event-based F1 scores, and empirically, we find that this also leads to near-optimal values in terms of the other considered evaluation metrics. For the encoder weight associated with acoustic information, we test the following options during training as well as inference: 0.25, 0.5, 0.75 and 0.875. When visuals are unavailable, this number is set to 1, as explained in Section 2.2.

Discussion on pretrained visual features A comparison between the results of the bi-encoder systems incorporating visual information and the scores obtained by the baseline models only utilizing acoustic inputs, both listed in Table 5, demonstrates that the proposed method can be useful, but the choice of pretrained vision-related vectors is crucial. Particularly, we observe that adding VGG16 embeddings does not globally lead to improvements, but only for clip- and segment-based F1 scores. However, the inclusion of OpenL3 and temporally coherent features does provide consistent and substantial performance boosts.

These findings are in agreement with outcomes published in prior research. In [5], it is shown that adding VGG16 embeddings can lead to improvements for audio tagging and (partly) coarse-grained sound event detection, but when it comes to stricter segmentation, these vectors do not add any value. This can be explained by the fact that the system these embeddings are extracted from is designed for a problem without time-related aspects, i.e., image classification. This is not the case for OpenL3 and temporally coherent visual features, as the associated models are pretrained for tasks encompassing temporal facets, which explains the disparity discussed in the previous paragraph.

Discussion on metrics Independent model initialization through random seeding of learnable parameters does not cause a lot of inconsistency with regard to the clip- and segment-based F1 scores, which measure audio tagging and coarse-grained sound event detection performance respectively. This allows us to conclude with certainty that systems employing the proposed method outperform the baseline when it comes to these metrics. For the event-based F1 measure, targeting comparatively strict temporal segmentation, the standard deviations across training runs listed in Table 5 are slightly bigger, but still small enough to be able to make useful inferences. However, the variability is significantly greater (in relative terms) for PSDS1 and especially PSDS2, which means more caution should be exercised when interpreting those results.

Discussion on interpolation weights We find that changing the encoder weights employed while training causes relatively limited fluctuations in terms of the considered metrics. However, modifying the inference hyperparameters can cause significant performance drops. This happens in particular when the interpolation weight for the acoustic stream is too low.

Based on this observation, we present a novel way of setting these hyperparameters. While learning, we randomize this decision process: The interpolation weight associated with the acoustic input features is sampled from a uniform distribution between 0.25 and 1 per batch. It would not be logical to let this value go all the way to 0, as in that specific case, the model would have to rely on visual information only. For reasons explained in detail in Section 1, this is not a good idea as the auditory stream is generally much more salient. For inference, the default procedure for determining these

Table 6 Results of bi-encoder attention-based models with random encoder interpolation weights during training

Base model	Encoder inputs	Encoder weight	CBF1	SBF1	EBF1	PSDS1	PSDS2
		Inference	(%)	(%)	(%)		
Transformer	Spectral auditory features	0.75	84.20	**80.71**	**50.80**	0.4169	**0.6557**
	OpenL3 visual features	0.25	±0.39	±0.48	±0.65	±0.0089	±0.010
	Spectral auditory features	0.875	84.42	80.46	50.64	**0.4201**	0.6554
	Temporally coherent visual features	0.125	±0.37	±0.39	±0.58	±0.0079	±0.013
	Spectral auditory features	0.875	**84.48**	80.20	46.95	0.3818	0.6170
	VGG16 features	0.125	±0.47	±0.39	±0.55	±0.0075	±0.014
Conformer	Spectral auditory features	0.875	83.94	80.27	**50.00**	**0.4051**	0.6539
	OpenL3 visual features	0.125	±0.39	±0.44	±0.64	±0.0074	±0.011
	Spectral auditory features	0.875	84.18	**80.50**	49.94	0.4040	**0.6552**
	Temporally coherent visual features	0.125	±0.36	±0.38	±0.57	±0.0078	±0.013
	Spectral auditory features	0.875	**84.25**	80.33	46.10	0.3591	0.6201
	VGG16 features	0.125	±0.39	±0.50	±0.67	±0.0089	±0.012

weights, involving optimization based on validation data, is retained. The results obtained when using this adapted method are presented in Table 6. The reported scores are very comparable to those in Table 5, for models that do not use randomized encoder interpolation weights during training. They are not much (or in some cases, any) better, but this altered learning procedure still has a significant benefit: The hyperparameter optimization process becomes less tedious and time-consuming in this case, which is valuable from a practical point of view.

Discussion on merit of multi-encoder learning In this section, we go into the merit of the multi-encoder learning framework by further scrutinizing the performance boosts reported in Table 5. To this end, we first split the test data into two sets, using the availability of visual information as partitioning criterion. We report the results obtained by the uni-encoder baseline as well as the proposed (non-randomized) bimodal models on these two disjoint collections. Additionally, we disclose performance scores produced by the latter systems on the evaluation data with visual assistance when the interpolation weight for the acoustic stream is forced to 1, and vision-related inputs are ignored. These outcomes are all given in Table 7. Standard deviations are not included to maintain clarity and conserve space.

Even though the bi-encoder attention-based architectures are designed to utilize inputs from two modalities, when vision-related inputs are forcibly ignored by setting the associated linear interpolation weights to 0, these models still perform on par to the uni-encoder baseline. This trend appears when inspecting the scores obtained on both of the aforementioned evaluation data splits. The only exception to this rule are the systems with pretrained VGG16 embeddings, but this is not entirely unexpected behavior, since these vectors are inappropriate for the tasks at hand as discussed before. This finding seems to indicate that, on the condition that the employed sequences of features are selected appropriately, the multi-encoder framework can be applied relatively safely, without fear of decreasing performance, regardless of the actual degree of accessibility of visual recordings.

When looking at the performance differences between the unimodal baseline and the bi-encoder models on the evaluation data that includes visual inputs, we find that there are consistently substantial improvements — once again, with the sole exception being systems utilizing pretrained VGG16 features. This is obviously desirable as the whole purpose of this research project is to be able to exploit multimodal inputs even when not all sources are always available.

Unfortunately, these findings also uncover the limitations of the proposed multi-encoder framework: In this instance, representing the important situation of data originating from YouTube, vision-related content is inaccessible for about 15.5% of all samples. For this case, the gains our proposed method is able to achieve are certainly worthwhile. If the amount of examples with visual assistance would diminish further, the performance boosts would also inevitably lessen, up to a point where it may no longer be worth the extra effort.

3.2.3 Tri-encoder attention-based models

Table 8 lists the scores obtained by models that use three encoders: One takes in spectral auditory maps, the two others accept OpenL3 and temporally coherent visual features respectively. Pretrained VGG16 embeddings are not investigated any further: As discussed at length in the previous sections, in the bi-encoder experimental configuration, these vectors have already been found to be unsuitable for the tasks at hand, particularly for fine-grained sound event detection.

Table 7 Split results of uni- and bi-encoder attention-based models

Base model	Encoder inputs	Evaluation data	CBF1 (%)	SBF1 (%)	EBF1 (%)	PSDS1	PSDS2
Transformer	Spectral auditory features	Video available (but ignored)	82.79	79.19	48.89	0.4026	**0.6377**
		Video unavailable	80.40	77.21	49.79	0.4358	0.6159
	Spectral auditory features OpenL3 visual features	Video available	85.10	80.94	50.70	0.4140	0.6583
		Video available (but ignored)	83.26	79.11	49.03	0.4073	0.6296
		Video unavailable	81.01	78.16	51.44	0.4311	0.6154
	Spectral auditory features Temporally coherent visual features	Video available	84.58	80.60	50.71	0.4163	0.6514
		Video available (but ignored)	82.74	78.98	49.44	0.4073	0.6248
		Video unavailable	80.91	77.03	51.41	0.4296	0.6099
	Spectral auditory features VGG16 features	Video available	85.16	80.73	48.34	0.3854	0.6222
		Video available (but ignored)	79.40	75.78	45.16	0.3591	0.5897
		Video unavailable	77.98	74.67	48.05	0.3926	0.5526
Conformer	Spectral auditory features	Video available (but ignored)	82.85	78.82	48.04	0.3811	0.6426
		Video unavailable	80.24	77.33	49.68	0.4210	0.6236
	Spectral auditory features OpenL3 visual features	Video available	84.35	80.70	50.14	0.3954	0.6477
		Video available (but ignored)	82.68	79.11	48.43	0.3878	0.6383
		Video unavailable	80.28	77.12	50.46	0.4256	0.6330
	Spectral auditory features Temporally coherent visual features	Video available	84.70	80.94	50.00	0.3970	0.6489
		Video available (but ignored)	82.38	78.83	48.28	0.3845	0.6370
		Video unavailable	80.45	77.50	50.87	0.4185	0.6284
	Spectral auditory features VGG16 features	Video available	84.03	80.19	46.09	0.3605	0.6290
		Video available (but ignored)	79.02	74.71	43.91	0.3486	0.5900
		Video unavailable	78.35	74.41	45.55	0.3714	0.5583

The interpolation hyperparameters are decided using procedures similar to those employed in the previously described experiments. As a first option, their values are chosen by optimizing event-based F1 scores on the validation partition of the considered data set. We test the following possibilities during training as well as inference for the auditory information weight: 0.25, 0.5, 0.75 and 0.875. Alternatively, the learning weights linked to the acoustic inputs are sampled randomly from a uniform distribution ranging from 0.25 to 1. To limit the search space in both of the foregoing cases, we choose to split the remaining share equally between the encoders connected to the visual streams.

When comparing results obtained by audio-only models (in Table 5) to those of bi-encoder systems also incorporating OpenL3 or temporally coherent visual embeddings (in Table 5 and Table 6), we mostly find consistent and substantial improvements. This has been discussed at length in the previous section(s). When inspecting the scores produced by tri-encoder architectures utilizing all three sets of relevant features (Table 8), we observe additional performance increases. However, these boosts are far less pronounced.

This makes sense as, in contrast to the initial switch from unimodal to bi-encoder models, we are not integrating a new modality into the systems, we are simply using more sets of (visual) features. It is very likely that much of the information captured in one series of vision-related embeddings is also present in the other, leading to diminishing returns in terms of performance when combining them. However, this experiment does demonstrate the flexibility of the proposed framework and how it could be employed when more than two groups of complementary features are available.

4 Conclusion

We proposed a dynamic multi-encoder approach to deal with the problem of missing values in the context of multimodal sound recognition. This particular situation frequently occurs since many pertinent data sets stem from YouTube, which provides a noteworthy opportunity but also poses a serious challenge: Visual inputs can be taken advantage of to enhance audio tagging and sound event detection models, which traditionally only employ acoustic information. However, vision-related features may not be accessible for all data points due to a variety of availability issues. We applied the aforementioned method to state-of-the-art attention-based neural network architectures.

We performed experiments using the data set associated with task 4 of the DCASE 2020 challenge and verified that the presented framework can lead to noteworthy performance boosts in a selection of different evaluation settings. We thoroughly investigated the outcomes of said trials and analyzed some properties and limitations of the introduced technique. Among other things, we showed that improvements were contingent on a good choice of (pretrained) visual features to be used in conjunction with spectral auditory maps and we demonstrated that the proposed method even holds up when all vision-related inputs are ignored.

The proposed framework is naturally flexible, and consequentially, there are some compelling possibilities for future research involving this principle. Firstly, further

Table 8 Results of tri-encoder attention-based models

Base model	Encoder inputs	Encoder weights		CBF1	SBF1	EBF1	PSDS1	PSDS2
		Training	Inference	(%)	(%)	(%)		
Transformer	Spectral auditory features	0.75	0.875	**85.00**	81.10	**52.01**	0.4194	**0.6653**
	OpenL3 visual features	0.125	0.0625	±0.37	±0.45	±0.63	±0.0091	±0.013
	Temporally coherent visual features	0.125	0.0625					
	Spectral auditory features	Random	0.75	84.86	**81.40**	51.82	**0.4231**	0.6622
	OpenL3 visual features	Random	0.125	±0.36	±0.40	±0.55	±0.0086	±0.011
	Temporally coherent visual features	Random	0.125					
Conformer	Spectral auditory features	0.75	0.875	**84.90**	**81.12**	**50.91**	**0.4097**	0.6612
	OpenL3 visual features	0.125	0.0625	±0.40	±0.49	±0.59	±0.0084	±0.012
	Temporally coherent visual features	0.125	0.0625					
	Spectral auditory features	Random	0.875	84.57	80.99	50.90	0.4087	**0.6647**
	OpenL3 visual features	Random	0.0625	±0.41	±0.41	±0.61	±0.0081	±0.012
	Temporally coherent visual features	Random	0.0625					

attention could be directed to the way encoder interpolation hyperparameters are set during training and inference: In this work, we started with fixed weights, optimized on held-out validation data. Afterwards, we showed that randomizing these values during training can lead to similar results and significantly less tuning. It could be interesting to investigate more complex schemes, such as data-dependent weighting. Secondly, other sets of features could be explored. Here, we stuck to utilizing pretrained visual features on top of rudimentary spectral auditory maps, but pretrained auditory features might be worthwhile as well. Lastly, this technique is in no way tied to sound recognition, and it could easily be applied to other research tasks which also encounter missing value problems.

Abbreviations

DCASE: Detection and Classification of Acoustic Scenes and Events; ReLU: Rectified linear unit; CBF1: Clip-based F1 score; SBF1: Segment-based F1 score; EBF1: Event-based F1 score; PSDS: Polyphonic sound event detection score.

Acknowledgements

This work was supported by a PhD Fellowship of Research Foundation Flanders (FWO-Vlaanderen) and the Flemish Government under "Onderzoeksprogramma AI Vlaanderen".

Author's Contributions

Wim Boes conducted the research and performed the experiments. Hugo Van hamme guided and supervised the project. All authors read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

The data set analyzed during the current study is available in the DCASE 2020 task 4 repository, http://dcase.community/challenge2020/task-sound-event-detection-and-separation-in-domestic-environments.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 4 April 2022 Accepted: 24 July 2022
Published online: 08 October 2022

References

1. F. Font, A. Mesaros, D.P.W. Ellis, E. Fonseca, M. Fuentes, B. Elizalde, Proceedings of the 6th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2021) (Universitat Pompeu Fabra, Spain, 2021)
2. S. Parekh, S. Essid, A. Ozerov, N.Q.K. Duong, P. Pérez, G. Richard, Weakly supervised representation learning for audio-visual scene analysis. IEEE/ACM Trans. Audio Speech Lang. Process. **28**, 416–428 (2019)
3. W. Boes, H. Van hamme, in *Proceedings of the 27th ACM International Conference on Multimedia*. Audiovisual transformer architectures for large-scale classification and synchronization of weakly labeled audio events (ACM, Nice, France, 2019), pp. 1961–1969
4. Y. Yin, H. Shrivastava, Y. Zhang, Z. Liu, R.R. Shah, R. Zimmermann, in *Proceedings of the AAAI Conference on Artificial Intelligence*. Enhanced audio tagging via multi-to single-modal teacher-student mutual learning (AAAI, Palo Alto, CA, USA, 2021), pp. 10709–10717
5. W. Boes, H. Van hamme, in *Proceedings of Interspeech 2021*. Audio-visual transfer learning for audio tagging and sound event detection (ISCA, Brno, Czechia, 2021), pp. 2401–2405
6. J.F. Gemmeke, D.P.W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R.C. Moore, M. Plakal, M. Ritter, in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Audio Set: An ontology and human-labeled dataset for audio events (IEEE, New Orleans, LA, USA, 2017), pp. 776–780
7. C.D. Kim, B. Kim, H. Lee, G. Kim, 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)*. AudioCaps: Generating captions for audios in the wild (ACL, Minneapolis, MN, USA, 2019), pp. 119–132
8. E. Tlamelo, M. Thabiso, M. Dimane, S. Thabo, M. Banyatsang, T. Oteng, A survey on missing data in machine learning. J. Big Data **8** (2021), pp. 1-37
9. D. Ramachandram, G.W. Taylor, Deep multimodal learning: A survey on recent advances and trends. IEEE Signal Proc Mag. **34**, 96–108 (2017)
10. T.D. Le, R. Beuran, Y. Tan, in *2018 10th International Conference on Knowledge and Systems Engineering*. Comparison of the most influential missing data imputation algorithms for healthcare (IEEE, Ho Chi Minh City, Vietnam, 2018), pp. 247–251
11. B.O. Petrazzini, H. Naya, F. Lopez-Bello, G. Vazquez, L. Spangenberg, Evaluation of different approaches for missing data imputation on features associated to genomic data. BioData mining. **14**, 1–13 (2021)
12. Y. Lee, S.-W. Kim, S. Park, X. Xie, in *Proceedings of the 2018 World Wide Web Conference*. How to impute missing ratings? Claims, solution, and its application to collaborative filtering (International World Wide Web Conferences Steering Committee, Lyon, France, 2018), pp. 783–792
13. K.E. Kafoori, S.M. Ahadi, Robust recognition of noisy speech through partial imputation of missing data. Circ Syst Sig Process. **37**, 1625–1648 (2018)

14. R.J.. Little, Little, Missing data assumptions. Ann Rev Stat Appl **8**, 89–107 (2021)
15. T. Lohrenz, Z. Li, T. Fingscheidt, in *Proceedings of Interspeech 2021*. Multi-encoder learning and stream fusion for transformer-based end-to-end automatic speech recognition (ISCA, Brno, Czechia, 2021), pp. 2846–2850
16. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, in *Advances in Neural Information Processing Systems*. Attention is all you need (NeurIPS, Long Beach, CA, USA, 2017), pp. 5998–6008
17. A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, R. Pang, in *Proceedings of Interspeech 2020*. Conformer: Convolution-augmented transformer for speech recognition (ISCA, Shanghai, China, 2020), pp. 5036–5040
18. R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, T. Liu, in *International Conference on Machine Learning*. On layer normalization in the transformer architecture (PMLR, Vienna, Austria, 2020), pp. 10524–10533
19. J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, in *Proceedings of NAACL-HLT*, BERT: Pre-training of deep bidirectional transformers for language understanding (ACL, Minneapolis, MN, USA, 2019) pp. 4171–4186
20. J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization. arXiv preprint arXiv:1607.06450. (2016)
21. K. He, X. Zhang, S. Ren, J. Sun, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Deep residual learning for image recognition (IEEE, Los Alamitos, CA, USA, 2016), pp. 770–778
22. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**, 1929–1958 (2014)
23. J. Gehring, M. Auli, D. Grangier, D. Yarats, Y.N. Dauphin, in *International Conference on Machine Learning*. Convolutional sequence to sequence learning (PMLR, Sydney, Australia, 2017), pp. 1243–1252
24. K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, K. Takeda, in *Proceedings of the 5th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2020)*. Conformer-based sound event detection with semi-supervised learning and data augmentation (Zenodo, Tokyo, Japan, 2020), pp. 100–104
25. Y. Lu, Z. Li, D. He, Z. Sun, B. Dong, T. Qin, L. Wang, T.-Y. Liu, in *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*. Understanding and improving transformer from a multi-particle dynamic system point of view (OpenReview, Addis Ababa, Ethiopia, 2020)
26. P. Ramachandran, B. Zoph, Q.V. Le, Searching for activation functions. arXiv preprint arXiv:1710.05941 (2017)
27. Z. Dai, Z. Yang, Y. Yang, J.G. Carbonell, Q. Le, R. Salakhutdinov, in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Transformer-xl: Attentive language models beyond a fixed-length context (ACL, Florence, Italy, 2019), pp. 2978–2988
28. Y.N. Dauphin, A. Fan, M. Auli, D. Grangier, in *International Conference on Machine Learning*. Language modeling with gated convolutional networks (PMLR, Sydney, Australia, 2017), pp. 933–941
29. S. Ioffe, C. Szegedy, in *International Conference on Machine Learning*. Batch normalization: Accelerating deep network training by reducing internal covariate shift (PMLR, Lille, France, 2015), pp. 448–456
30. N. Turpault, R. Serizel, A. Parag Shah, J. Salamon, in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*. Sound event detection in domestic environments with weakly labeled data and soundscape synthesis (New York University, New York, NY, USA, 2019), pp. 253–257
31. O. Nobutaka, H. Noboru, K. Yohei, M. Annamaria, I. Keisuke, K. Yuma, K. Tatsuya, Proceedings of the 5th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2020) (Zenodo, Japan, 2020)
32. Turpault, N. Serizel, R. in *Proceedings of the 5th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2020)*. Training sound event detection on a heterogeneous dataset (Zenodo, Tokyo, Japan, 2020), pp. 200–204
33. J. Cramer, H.-H. Wu, J. Salamon, J.P. Bello, in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Look, listen, and learn more: Design choices for deep audio embeddings (IEEE, Brighton, UK, 2019), pp. 3852–3856
34. J. Knights, B. Harwood, D. Ward, A. Vanderkop, O. Mackenzie-Ross, P. Moghadam, in *2020 25th International Conference on Pattern Recognition (ICPR)*. Temporally coherent embeddings for self-supervised video representation learning (IEEE, Milan, Italy, 2021), pp. 8914–8921
35. K. Simonyan, A. Zisserman, in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Very deep convolutional networks for large-scale image recognition (OpenReview, San Diego, CA, USA, 2015)
36. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge. Int. J. Comput. Vis. (IJCV) **115**, 211–252 (2015)
37. A. Mesaros, T. Heittola, T. Virtanen, Metrics for polyphonic sound event detection. Appl. Sci. **6** (2016), p. 162
38. T. Virtanen, A. Mesaros, T. Heittola, A. Diment, E. Vincent, E. Benetos, B.M. Elizalde, Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017) (Tampere University of Technology, Germany, 2017)
39. M.D. Plumbley, C. Kroos, J.P. Bello, G. Richard, D.P.W. Ellis, A. Mesaros, Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018) (Tampere University of Technology, United Kingdom, 2018)
40. M. Mandel, J. Salamon, D.P.W. Ellis, Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019) (New York, United States of America, 2019)
41. Ç. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, S. Krstulović, in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. A framework for the robust evaluation of sound event detection (IEEE, Barcelona, Spain, 2020), pp. 61–65
42. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, in *Advances in Neural Information Processing Systems*. PyTorch: An imperative style, high-performance deep learning library (NeurIPS, Vancouver, Canada, 2019), pp. 8026–8037
43. A. Tarvainen, H. Valpola, in *Advances in Neural Information Processing Systems*. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results (NeurIPS, Long Beach, CA, 2017), pp.1195–1204
44. H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz, in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30-May 3, 2018, Conference Track Proceedings*. mixup: Beyond empirical risk minimization (PMLR, Vancouver, Canada, 2018)
45. D.P. Kingma, J. Ba, in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Adam: A method for stochastic optimization (PMLR, San Diego, CA, USA, 2015)
46. L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, J. Han, in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, Conference Track Proceedings*. On the variance of the adaptive learning rate and beyond (PMLR, New Orleans, LA, USA, 2019)

Publisher's Note