Localizing Visual Sounds the Hard Way

Honglie Chen, Weidi Xie, Triantafyllos Afouras, Arsha Nagrani, Andrea Vedaldi, Andrew Zisserman
VGG, Department of Engineering Science, University of Oxford, UK
{hchen, weidi, afourast, arsha, vedaldi, az}@robots.ox.ac.uk

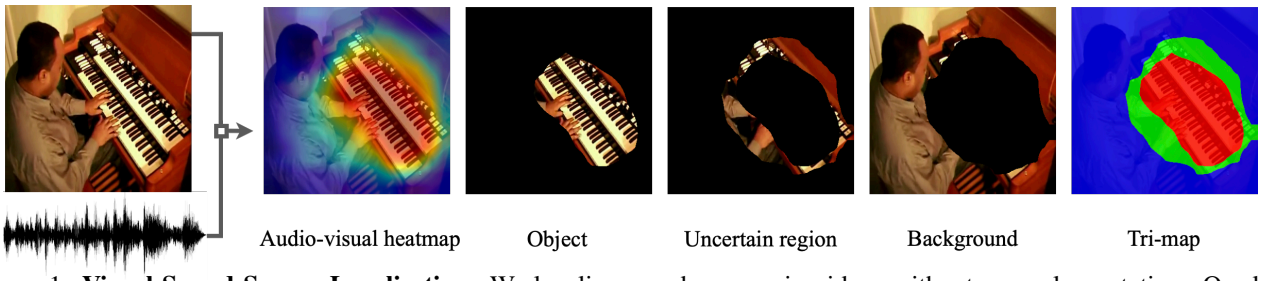


Figure 1: **Visual Sound Source Localisation:** We localise sound sources in videos without manual annotation. Our key contribution is an automatic negative mining technique through differentiable thresholding of a cross-modal correspondence score map into a Tri-map. We use background regions with low correlation to the given sound as 'hard negatives' in a contrastive learning framework.

Abstract

The objective of this work is to localize sound sources that are visible in a video without using manual annotations. Our key technical contribution is to show that, by training the network to explicitly discriminate challenging image fragments, even for images that do contain the object emitting the sound, we can significantly boost the localization performance. We do so elegantly by introducing a mechanism to mine hard samples and add them to a contrastive learning formulation automatically. We show that our algorithm achieves state-of-the-art performance on the popular Flickr SoundNet dataset. Furthermore, we introduce the VGG-Sound Source (VGG-SS) benchmark, a new set of annotations for the recently-introduced VGG-Sound dataset, where the sound sources visible in each video clip are explicitly marked with bounding box annotations. This dataset is 20 times larger than analogous existing ones, contains 5K videos spanning over 200 categories, and, differently from Flickr SoundNet, is video-based. On VGG-SS, we also show that our algorithm achieves state-of-the-art performance against several baselines. Code and datasets can be found at http://www.robots.ox.ac.uk/~vgg/research/lvs/.

1. Introduction

While research in computer vision largely focuses on the visual aspects of perception, natural objects are characterized by much more than just appearance. Most objects, in particular, emit sounds, either in their own right, or in their interaction with the environment — think of the bark of a dog, or the characteristic sound of a hammer striking a nail. A full understanding of natural objects should not ignore their acoustic characteristics. Instead, modelling appearance and acoustics jointly can often help us understand them better and more efficiently. For example, several authors have shown that it is possible to use sound to discover and localize objects automatically in videos, without the use of any manual supervision [1, 2, 14, 17, 24, 30].

In this paper, we consider the problem of localizing 'visual sounds', *i.e.* visual objects that emit characteristics sounds in videos. Inspired by prior works [2, 14, 30], we formulate this as finding the correlation between the visual and audio streams in videos. These papers have shown that not only can this correlation be learned successfully, but that, once this is done, the resulting convolutional neural networks can be 'dissected' to localize the sound source spatially, thus imputing it to a specific object. However, other than in the design of the architecture itself, there is little in this prior work meant to improve the localization capabilities of the resulting models. In particular, while several models [1, 2, 30] do incorporate a form of spatial attention which should also help to localize the sounding object as a byproduct, these may still fail to provide a good *coverage* of the object, often detecting too little or too much of it.

In order to address this issue, we propose a new training scheme that explicitly seeks to spatially localize sounds in

video frames. Similar to object detection [35], in most cases only a small region in the image contains an object of interest, in our case a 'sounding' object, with the majority of the image often being 'background' which is not linked to the sound. Learning accurate object detectors involves explicitly seeking for these background regions, prioritizing those that could be easily confused for the object of interest, also called *hard negatives* [7, 13, 21, 28, 31, 35]. Given that we lack supervision for the location of the object making the sound, however, we are unable to tell which boxes are positive or negative. Furthermore, since we seek to solve the localization rather than the detection problem, we do not even have bounding boxes to work with, as we seek instead a segmentation of the relevant image area.

In order to incorporate hard evidence in our unsupervised (or self-supervised) setting, we propose an automatic background mining technique through differentiable thresholding, *i.e*. regions with low correlation to the given sound are incorporated into a negatives set for contrastive learning. Instead of using hard boundaries, we note that some regions may be uncertain, and hence we introduce the concept of a Tri-map into the training procedure, leaving an 'ignore' zone for our model. To our knowledge, this is the first time that background regions have been explicitly considered when solving the sound source localization problem. We show that this simple change significantly boosts sound localization performance on standard benchmarks, such as Flickr SoundNet [30].

To further assess sound localization algorithms, we also introduce a new benchmark, based on the recently-introduced VGG-Sound dataset [4], where we provide high-quality bounding box annotations for 'sounding' objects, *i.e*. objects that produce a sound, for more than 5K videos spanning 200 different categories. This dataset is 20$\times$ larger and more diverse than existing sound localization benchmarks, such as Flickr SoundNet (the latter is also based on still images rather than videos). We believe this new benchmark, which we call VGG-Sound Source, or VGG-SS for short, will be useful for further research in this area. In the experiments, we establish several baselines on this dataset, and further demonstrate the benefits of our new algorithm.

2. Related Work

2.1. Audio-Visual Sound Source Localization

Learning to localize sound sources by exploiting the natural co-occurrence of visual and audio cues in videos has a long history. Early attempts to solve the task used shallow probabilistic models [9, 16, 20], or proposed segmenting videos into spatio-temporal tubes and associating those to the audio signal through canonical correlation analysis (CCA) [18].

Modern approaches solve the problem using deep neural networks — typically employing a dual stream, trained with a contrastive loss by exploiting the audio-visual correspondence, *i.e*. matching audio and visual representations extracted from the same video. For example, [2, 14, 27, 30] associate the appearance of objects with their characteristic sounds or audio narrations; Hu *et al*. [17] first cluster audio and visual representations within each modality, followed by associating the resulting centroids with contrastive learning; Qian *et al*. [26] proposed a weakly supervised approach, where the approximate locations of the objects are obtained from CAMs to bootstrap the model training. Apart from using correspondence, Owens and Efros [25] also localize sound sources through synchronization, a related objective also investigated in earlier works [6, 22], while [19] incorporate explicit attention in this model. Afouras *et al*. [1] also exploit audio-visual concurrency to train a video model that can distinguish and group instances of the same category.

Alternative approaches solve the task using an audio-visual source separation objective. For example Zhao *et al*. [38] employ a mix-and-separate approach to learn to associate pixels in video frames with separated audio sources, while Zhao *et al*. [37] extends this method by providing the model with motion information through optical flow. Rouditchenko *et al*. [29] train a two-stream model to co-segment video and audio, producing heatmaps that roughly highlight the object according to the audio semantics. These methods rely on the availability of videos containing single-sound sources, usually found in well curated datasets. In other related work, Gan *et al*. [10] learn to detect cars from stereo sound, by distilling video object detectors, while Gao *et al*. [11] lift mono sound to stereo by leveraging spatial information.

2.2. Audio-Visual Localization Benchmarks

Existing audio-visual localization benchmarks are summarised in Table 1 (focusing on the test sets). The Flickr SoundNet sound source localization benchmark [30] is an annotated collection of single frames randomly sampled from videos of the Flickr SoundNet dataset [3, 33]. It is currently the standard benchmark for the sound source localization task; we discuss its limitations in Section 4, where we introduce our new benchmark. The Audio-Visual Event (AVE) dataset [34], contains 4,143 10 second video clips spanning 28 audio-visual event categories with temporal boundary annotations. LLP [36] contains of 11,849 YouTube video clips spanning 25 categories for a total of 32.9 hours collected from AudioSet [12]. The development set is sparsely annotated with object labels, while the test set contains dense video and audio sound event labels on the frame level. Note that the AVE and LLP test sets contain only temporal localisation of sounds (at the frame level),

with no spatial bounding box annotation.

Benchmark Datasets	# Data	# Classes	Video	BBox
Flickr SoundNet [30]	250	∼50‡	×	✓
AVE [34]†	402	28	✓	×
LLP [36]†	1,200	25	✓	×
VGG-SS	5,158	220	✓	✓

Table 1: Comparison with the existing sound-source localisation benchmrks. Note that VGG-SS has more images and classes. †These datasets contain only temporal localisation of sounds, not spatial localisation. ‡ We determined this via manual inspection.

3. Method

Our goal is to localize objects that make characteristic sounds in videos, without using any manual annotation. Similar to prior work [2], we use a two-stream network to extract visual and audio representations from unlabelled video. For localization, we compute the cosine similarity between the audio representation and the visual representations extracted convolutionally at different spatial locations in the images. In this manner, we obtain a positive signal that pulls together sounds and relevant spatial locations. For learning, we also need an opposite negative signal. A weak one is obtained by correlating the sound to locations in other, likely irrelevant videos. Compared to prior work [1, 2], our key contribution is to *also* explicitly seek for hard negative locations that contain background or non-sounding objects in the *same* images that contain the sounding ones, leading to more selective and thus precise localization. An overview of our architecture can be found in Figure 2.

While the idea of using hard negatives is intuitive, an effective implementation is less trivial. In fact, while we seek for hard negatives, there is no hard evidence for whether any region is in fact positive (sounding) or negative (non-sounding) as videos are unlabelled. An incorrect classification of a region as positive or negative can throw off the localization algorithm entirely. We solve this problem by using a robust contrastive framework that combines soft thresholding and Tri-maps, which enables us to handle uncertain regions effectively.

In sections 3.1 to 3.3 we first describe the task of audio-visual localization using contrastive learning in its *oracle* setting, assuming, for each visual-audio pair, we do have the ground-truth annotation for which region in the image is emitting the sound. In section 3.4, we introduce our proposed idea, which replaces the *oracle*, and discuss the difference between our method and existing approaches.

3.1. Audio-Visual Feature Representation

Given a short video clip with N visual frames and audio, and considering the center frame as visual input, *i.e.* $X = \{I, a\}, I \in \mathbb{R}^{3\times H_v \times W_v}, a \in \mathbb{R}^{1\times H_a \times W_a}$. Here, I refers to the visual frame, and a to the spectrogram of the raw audio waveform. In this manner, representations for both modalities can be computed by means of CNNs, which we denote respectively $f(\cdot;\theta_1)$ and $g(\cdot;\theta_2)$. For each video X_i, we obtain visual and audio representations:

$$V_i = f(I_i;\theta_1), \quad V_i \in \mathbb{R}^{c\times h\times w}, \tag{1}$$

$$A_i = g(a_i;\theta_2), \quad A_i \in \mathbb{R}^{c}. \tag{2}$$

Note that both visual and audio representation have the same number of channels c, which allows to compare them by using dot product or cosine similarity. However, the video representation also has a spatial extent $h \times w$, which is essential for spatial localization.

3.2. Audio-Visual Correspondence

Given the video and audio representations of eqs. (1) and (2), we put in correspondence the audio of clip i with the image of clip j by computing the cosine similarity of the representations, using the audio as a probe vector:

$$[S_{i\to j}]_{uv} = \frac{\langle A_i, [V_j]_{:uv}\rangle}{\|A_i\|\ \|[V_i]_{:uv}\|}, \quad uv \in [h]\times[w].$$

This results in a map $S_{i\to j} \in \mathbb{R}^{h\times w}$ indicating how strongly each image location in clip j responds to the audio in clip i. To compute the cosine similarity, the visual and audio features are L^2 normalized. Note that we are often interested in correlating images and audio from the same clip, which is captured by setting $j = i$.

3.3. Audio-Visual Localization with an Oracle

In the literature, training models for audio-visual localization has been treated as learning the correspondence between these two signals, and formulated as contrastive learning [1, 2, 17, 26, 30].

Here, before diving into the self-supervised approach, we first consider the *oracle* setting for the contrastive learning where ground-truth annotations are available. This means that we are given a training set $\mathcal{D} = \{d_1, d_2, \ldots, d_k\}$, where each training sample $d_i = (X_i, m_i)$ consists of a audio-visual sample X_i, as given above, plus a segmentation mask $m_i \in \mathbb{B}^{h\times w}$ with ones for those spatial locations that overlap with the object that emits the sounds, and zeros elsewhere. During training, the goal is therefore to jointly optimize $f(\cdot;\theta_1)$ and $g(\cdot;\theta_2)$, such that $S_{i\to i}$ gives high responses only for the region that emits the sound present in the audio. In this paper, we consider a specific type of contrastive learning, namely, InfoNCE [23].

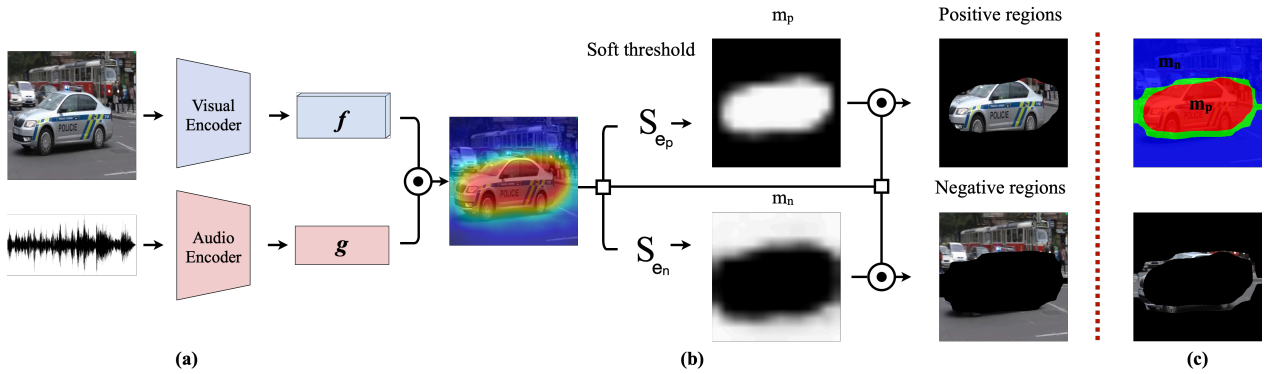


Figure 2: **Architecture Overview**. We use an audio-visual pair as input to a dual-stream network shown in **(a)**, $f(\cdot;\theta_1)$ and $g(\cdot;\theta_2)$, denoting the visual and audio feature extractor respectively. Cosine similarity between the audio vector and visual feature map is then computed, giving us a heatmap of size 14×14. **(b)** demonstrates the soft threshold being applied twice with different parameters, generating positive, negative regions. The final Tri-map and the uncertain region are highlighed in **(c)**.

Optimization. For each clip i in the dataset (or batch), we define the positive and negative responses as:

$$P_i = \frac{1}{|m_i|}\langle m_i,\ S_{i\to i}\rangle,$$
$$N_i = \underbrace{\frac{1}{|\mathbf{1}-m_i|}\langle \mathbf{1}-m_i,\ S_{i\to i}\rangle}_{\text{hard negatives}} + \underbrace{\frac{1}{hw}\sum_{i\neq j}\langle \mathbf{1},\ S_{i\to j}\rangle}_{\text{easy negatives}}.$$

where $\langle\cdot,\cdot\rangle$ denotes Frobenius inner product. To interpret this equation, note that the inner product simply sums over the element-wise product of the specified tensors and that $\mathbf{1}$ denotes a $h \times w$ tensor of all ones. The first term in the expression for N_i refers to the *hard negatives*, calculated from the "background" (regions that do not emit the characteristic sound) within the same image, and the second term denotes the easy negatives, coming from other images in the dataset. The optimization objective can therefore be defined as:

$$\mathcal{L} = -\frac{1}{k}\sum_{i=1}^{k}\left[\log\frac{\exp(P_i)}{\exp(P_i)+\exp(N_i)}\right]$$

Discussion. Several existing approaches [1, 2, 14, 30] to self-supervised audio-visual localization are similar. The key difference lies in the way of constructing the positive and negative sets. For example, in [30] a heatmap generated by using the soft-max operator is used to pool the positives and images from other video clips are treated as negatives; instead, in [2], positives come from max pooling the correspondence map, $S_{i\to i}$ and the negatives from max pooling $S_{i\to j}$ for $j \neq i$. Crucially, all such approaches have missed the *hard negatives* term defined above, computed from the background regions within the same images that do contain the sound. Intuitively this term is important to obtain a shaper visual localization of the sound source; however, while this is easy to implement in the oracle setting, obtaining hard negatives in self-supervised training requires some care, as discussed next.

3.4. Self-supervised Audio-Visual Localization

In this section, we describe a simple approach for replacing the oracle, and continuously bootstrapping the model to achieve better localization results. At a high level, the proposed idea inherits the spirit of self-training, where predictions are treated as pseudo-ground-truth for re-training.

Specifically, given a dataset $\mathcal{D} = \{X_1, X_2, \ldots, X_k\}$ where only audio-visual pairs are available (but not the masks m_i), the correspondence map $S_{i\to i}$ between audio and visual input can be computed in the same manner as section 3.2. To get the pseudo-ground-truth mask $\hat{m}_i$, we could simply threshold the map $S_{i\to i}$:

$$\hat{m}_i = \begin{cases} 1, & \text{if } S_{i\to i} \geq \epsilon \\ 0, & \text{otherwise} \end{cases}$$

Clearly, however, this thresholding, which uses the Heaviside function, is not differentiable. Next, we address this issue by relaxing the thresholding operator.

Smoothing the Heaviside function. Here, we adopt a smoothed thresholding operator in order to maintain the end-to-end differentiability of the architecture:

$$\hat{m}_i = \text{sigmoid}((S_{i\to i} - \epsilon)/\tau)$$

where ϵ refers to the thresholding parameter, and τ denotes the temperature controlling the sharpness.

Handling uncertain regions. Unlike the oracle setting, the pseudo-ground-truth obtained from the model prediction may potentially be noisy, we therefore propose to set up

an "ignore" zone between the positive and negative regions, allowing the model to self-tune. In the image segmentation literature, this is often called a Tri-map and is also used for matting [5, 32]. Conveniently, this can be implemented by applying two different ϵ's, one controlling the threshold for the positive part and the other for the negative part of the Tri-map.

Training objective. We are now able to replace the oracle while computing the positives and negatives automatically. This leads to our final formulation:

$$\hat{m}_{ip} = \text{sigmoid}((S_{i\to i} - \epsilon_p)/\tau)$$
$$\hat{m}_{in} = \text{sigmoid}((S_{i\to i} - \epsilon_n)/\tau)$$
$$P_i = \frac{1}{|\hat{m}_{ip}|}\langle \hat{m}_{ip},\ S_{i\to i}\rangle$$
$$N_i = \frac{1}{|\mathbf{1} - \hat{m}_{in}|}\langle \mathbf{1} - \hat{m}_{in},\ S_{i\to i}\rangle + \frac{1}{hw}\sum_{j\neq i}\langle \mathbf{1},\ S_{i\to j}\rangle$$
$$\mathcal{L} = -\frac{1}{k}\sum_{i=1}^{k}\left[\log\frac{\exp(P_i)}{\exp(P_i)+\exp(N_i)}\right]$$

where ϵ_p and ϵ_n are two thresholding parameters (validated in experiment section), with $\epsilon_p > \epsilon_n$. For example if we set $\epsilon_p = 0.6$ and $\epsilon_n = 0.4$, regions with correspondence scores above 0.6 are considered positive and bellow 0.4 negative, while the areas falling within the $[0.4, 0.6]$ range are treated as "uncertain" regions and ignored during training (Figure 2).

4. The VGG-Sound Source Benchmark

As mentioned in Section 2, the SoundNet-Flickr sound source localization benchmark [30] is commonly used for evaluation in this task. However, we found it to be unsatisfactory in the following aspects: i) both the number of total instances (250) and sounding object categories (approximately 50) that it contains are limited, ii) only certain reference frames are provided, instead of the whole video clip, which renders it unsuitable for the evaluation of video models, and iii) it provides no object category annotations.

In order to address these shortcomings, we build on the recent VGG-Sound dataset [4] and introduce VGG-SS, an audio-visual localization benchmark based on videos collected from YouTube.

4.1. Test Set Annotation Pipeline

In the following sections, we describe a semi-automatic procedure to annotate the objects that emit sounds with bounding boxes, which we apply to obtain VGG-SS with over 5k video clips, spanning 220 classes.

(1) Automatic bbox generation. We use the entire VGG-Sound test set, containing 15k 10-second video clips, and extract the center frame from each clip. We use a Faster R-CNN object detector [28] pretrained on OpenImages to predict the bounding boxes of all relevant objects. Following [4], we use a word2vec model to match visual and audio categories that are semantically similar. At this stage, there are roughly 8k frames annotated automatically.

(2) Manual image annotation. We then annotate the remaining frames manually. There are three main challenges at this point: (i) there are cases where localization is extremely difficult or impossible, either because the object is not visible (e.g. in extreme lighting conditions), too small ('mosquito buzzing'), or is diffused throughout the frame ('hail', 'sea waves', 'wind'); (ii) the sound may originate either from a single object, or from the interactions between multiple objects and a consistent annotation scheme must be decided upon; and finally (iii), there could be multiple instances of the same class in the same frame, and it is challenging to know which of the instances are making the sound from a single image.

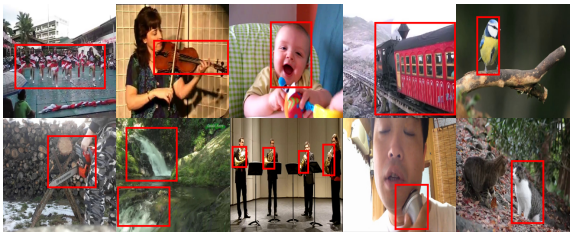
We address these issues in three ways: First, we remove categories (*e.g.* mainly environmental sounds such as wind, hail etc) that are challenging to localize, roughly 50 classes; Second, as illustrated in Figure 3a, when the sound comes from the interaction of multiple objects, we annotate a tight region surrounding the interaction point; Third, if there are multiple instances of the same sounding object category in the frame, we annotate each separately when there are less than 5 instances and they are separable, otherwise a single bounding box is drawn over the entire region, as shown in the top left image ('human crowd') in Figure 3a.

(3) Manual video verification. Finally, we conduct manual verification on videos using the VIA software [8]. We do this by watching the 5-second video around every annotated frame, to ensure that the sound corresponds with the object in the bounding box. This is particularly important for the cases where there are multiple candidate instances present in the frame, however, only one is making the sound, *e.g.* human singing.

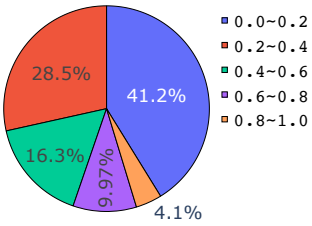
The statistics after every stage of the process and the final dataset are summarised in Table 2. The first stage generates bounding box candidates for the entire VGG-Sound test set (309 classes, 15k frames); the manual annotation process then removes unclear classes and frames, resulting in roughly 260 classes and 8k frames. Our final video verification further cleans up the the test set, yielding a high-quality large-scale audio-visual benchmark — VGG-Sound Source (VGG-SS), which is 20 times larger than the existing one [30].

5. Experiments

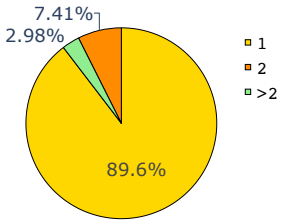
In the following sections, we describe the datasets, evaluation protocol and experimental details used to thoroughly



(a) VGG-SS benchmark examples



(b) Bounding box areas



(c) Number of bounding boxes

Figure 3: **VGG-SS Statistics.** Figure 3a: Example VGG-SS images and annotations showing class diversity (humans, animals, vehicles, tools etc.) Figure 3b: Distribution of bounding box areas in VGG-SS, the majority of boxes cover less than 40% of the image area. Figure 3c shows the distribution of number of bounding boxes - roughly 10% of the test data is challenging with more than one bounding box per image.

Stage	Goal	# Classes	# Videos
1	Automatic BBox Generation	309	15k
2	Manual Annotation	260	8k
3	Video Verification	220	5k

Table 2: The number of classes and videos in VGG-SS after each annotation stage.

assess our method.

5.1. Training Data

For training our models, we consider two large-scale audio-visual datasets, the widely used Flickr SoundNet dataset and the recent VGG-Sound dataset, as detailed next. Only the center frames of the *raw* videos are used for training. Note, other frames *e.g.* (3/4 of the video) are tried for training, no considerable performance change is observed.

Flickr SoundNet: This dataset was initially proposed in [3] and contains over 2 million unconstrained videos from Flickr. For a fair comparison with recent work [17, 26, 30], we follow the same data splits, conducting self-supervised training with subsets of 10k or 144k image and audio pairs.

VGG-Sound: VGG-Sound was recently released with over 200k clips for 300 different sound categories. The dataset is conveniently audio-visual, in the sense that the object that emits sound is often visible in the corresponding video clip, which naturally suits the task considered in this paper. Again, to draw fair comparisons, we conduct experiments with training sets consisting of image and audio pairs of varying sizes, *i.e.* 10k, 144k and the full set.

5.2. Evaluation protocol

In order to quantitatively evaluate the proposed approach, we adopt the evaluation metrics used in [26, 30]: Consensus Intersection over Union (cIoU) and Area Under Curve (AUC) are reported for each model on two test sets, as detailed next.

Flickr SoundNet Testset: Following [17, 26, 30], we report performance on the 250 annotated image-audio pairs of the Flickr SoundNet benchmark. Every frame in this test set is accompanied by 20 seconds of audio, centered around it, and is annotated with 3 separate bounding boxes indicating the location of the sound source, each performed by a different annotator.

VGG-Sound Source (VGG-SS): We also re-implement and train several baselines on VGG-Sound and evaluate them on our proposed VGG-SS benchmark, described in section 4.

5.3. Implementation details

As Flickr SoundNet consists of image-audio pairs, while VGG-Sound contains short video clips, when training on the latter we select the middle frame of the video clip and extract a 3s audio segment around it to create an equivalent image-audio pair. Audio inputs are 257×300 magnitude spectrograms. The dimensions for the audio output from the audio encoder CNN is a 512D vector, which is max-pooled from a feature map of $17 \times 13 \times 512$, where 17 and 13 refer to the frequency and time dimension respectively. For the visual input, we resize the image to a $224 \times 224 \times 3$ tensor without cropping. For both the visual and audio stream, we use a lightweight ResNet18 [15] as a backbone. Following the baselines [17, 26], we also pretrain the visual encoder on ImageNet. We use $\epsilon_p = 0.65$ and $\epsilon_n = 0.4$, $\tau = 0.03$. All models are trained with the Adam optimizer using a learning rate of 10^{-4} and a batch size of 256. During testing, we directly feed the full length audio spectrogram into the network.

6. Results

In the following sections, we first compare our results with recent work on both Flickr SoundNet and VGG-SS dataset in detail. Then we conduct an ablation analysis showing the importance of the *hard negatives* and the Tri-map in self-supervised audio-visual localization.

6.1. Comparison on the Flickr SoundNet Test Set

In this section, we compare to recent approaches by training on the same amount of data (using various different datasets). As shown in Table 3, we first fix the training set to be Flickr SoundNet with 10k training samples and compare our method with [2, 14, 26]. Our approach clearly outperforms the best previous methods by a substantial gap (0.546% vs. 0.582%). Second, we also train on VGG-Sound using 10k random samples, which shows the benefit of using VGG-Sound for training. Third, we switch to a larger training set consisting of 144k samples, which gives us a further 5% improvement compared to the previous state-of-the-art method [17]. In order to tease apart the effect of various factors in our proposed approach, *i.e*. introducing *hard negative* and using a Tri-map vs different training sets, *i.e*. Flickr144k vs. VGG-Sound144k, we conduct an ablation study, as described next.

Method	Training set	CIoU	AUC
Attention10k [30]	Flickr10k	0.436	0.449
CoarsetoFine [26]	Flickr10k	0.522	0.496
AVObject [1]	Flickr10k	0.546	0.504
Ours	Flickr10k	**0.582**	**0.525**
Ours	VGG-Sound10k	**0.618**	**0.536**
Attention10k [30]	Flickr144k	0.660	0.558
DMC [17]	Flickr144k	0.671	0.568
Ours	Flickr144k	**0.699**	**0.573**
Ours	VGG-Sound144k	**0.719**	**0.582**
Ours	VGG-Sound Full	**0.735**	**0.590**

Table 3: Quantitative results on Flickr SoundNet testset. We outperform all recent works using different training sets and number of training data.

Model	Pos ϵ	Neg ϵ	Tri-map	CIoU	AUC
a	✓	×	×	0.675	0.568
b	✓	✓	×	0.667	0.544
c	✓	✓	✓	0.719	0.582

Table 4: Method ablations. The amount of hard negatives are investigated here, only proper amount of negatives can benefit the models.

6.2. Ablation Analysis

In this section, we train our method using the 144k-samples training data from VGG-Sound and evaluate it on the Flickr SoundNet test set. The goal is to investigate the benefit of introducing *hard negative* regions and the Tri-map in the self-supervised learning formulation. As shown in table 4, we first note that using hard negatives naïvely does not help: comparing **model a** trained using only positives and **model b** adding negatives from the complementary region decreases performance slightly. This is because all the non-positive areas have been counted as negatives, whereas regions around the object are often hard to define. Therefore deciding for all pixels whether they are positive or negative is problematic. Second, comparing **model b** and **model c** where some areas between positives and negatives are ignored during training by using the Tri-map, we obtain a large 4.4% gain, demonstrating the importance of defining an "uncertain" region and allowing the model to self-tune. We show more results in the extended Arxiv version.

Method	CIoU	AUC
Attention10k [30]	0.185	0.302
AVobject [1]	0.297	0.357
Ours	**0.344**	**0.382**

Table 5: Quantitative results on the VGG-SS testset. All models are trained on VGG-Sound 144k and tested on VGG-SS.

6.3. Comparison on VGG-Sound Source

In this section, we evaluate the models on the newly proposed VGG-SS benchmark. As shown in Table 5, the CIoU is reduced significantly for all models compared to the results in Table 3, showing that VGG-SS is a more diverse and challenging benchmark than Flickr SoundNet. However, our proposed method still outperforms all other baseline methods by a large margin of around 5%.

6.4. Qualitative results

In Figure 4, we threshold the heatmaps with different thresholds, *e.g*. $\epsilon_p = 0.65$ and $\epsilon_n = 0.4$ (same as the ones used during training). The objects and background are accurately highlighted in the positive region and negative region respectively, so that the model can learn proper amount of hard negatives. We visualize the prediction results in Figure 5, and note that the proposed method presents much cleaner heatmap outputs. This once again indicates the benefits of considering hard negatives during training.

6.5. Open Set Audio-visual Localization

We have so far trained and tested our models on data containing the same sound categories (closed set classification). In this section we determine if our model trained on heard/seen categories can generalize to classes that have never been heard/seen before, *i.e*. to an open set scenario. To test this, we randomly sample 110 categories (seen/heard) from VGG-Sound for training, and evaluate our network on another *disjoint* set of 110 unseen/unheard categories (for a full list please refer to sup-

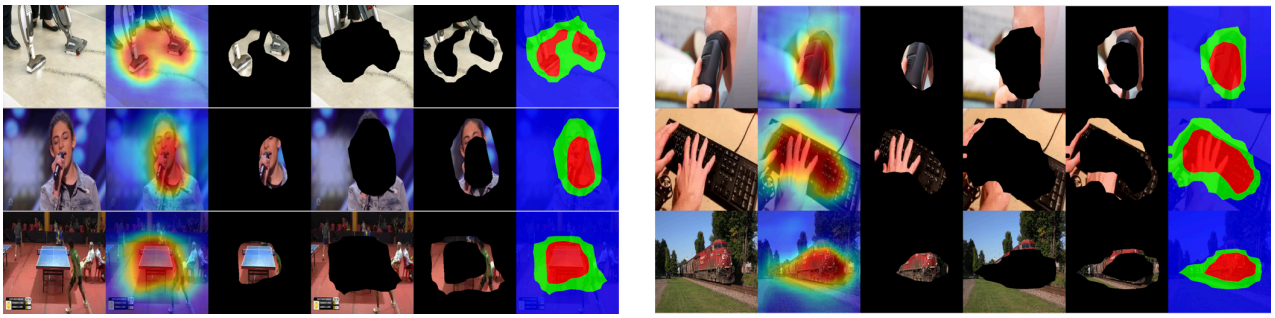
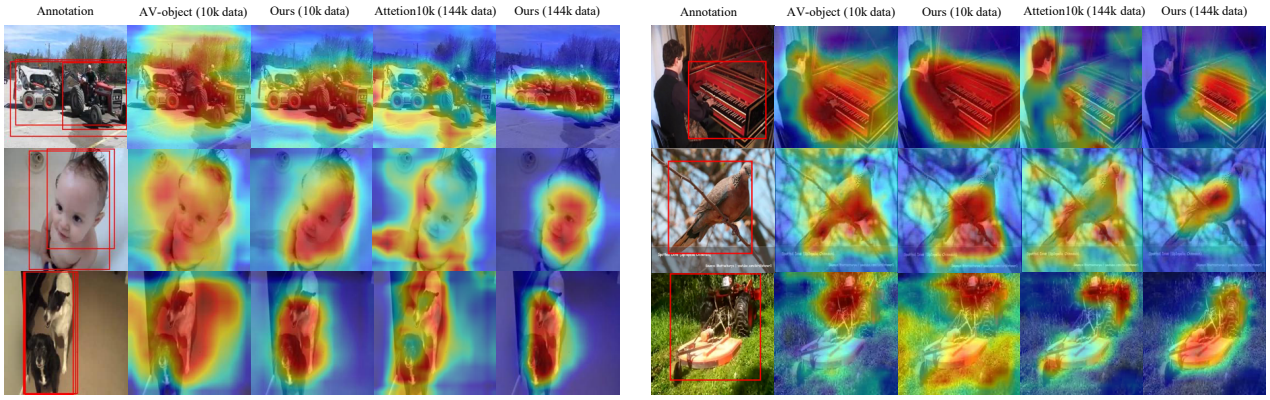


Figure 4: **Example Tri-map visualisations.** We show images, heatmaps and Tri-maps here. The Tri-map effectively identify the objects and the uncertain region let the model only learn controlled hard negatives.



(a) Visualisation on Flickr SoundNet testset

(b) Visualisation on VGG-SS testset

Figure 5: **Qualitative results** for models trained on various methods and data amount. The first column shows annotation overlaid on images, the following two column shows predictions trained on 10k data and the last tow column show predictions trained on 144k data. Our method has no false positives in the predictions as the hard negatives are penalised in the training.

# training Data	Test class	CIoU	AUC
70k	Heard 110	0.289	0.362
70k	Unheard 110	0.263	0.347

Table 6: Quantitative results on VGG-SS for unheard classes. We vary the training set (classes) and keep the testing set fixed (subset of the VGG-SS).

plementary). We use roughly 70k samples for both heard and unheard classes.

Heard and unheard evaluations are shown in Table 6, where for the heard split we also train the model on 70k samples containing both old and new classes. The difference in performance is only 2%, which demonstrates the ability of our network to generalize to unheard or unseen categories. This is not surprising due to the similarity between several categories. For example, if the training corpus contains human speech, one would expect the model to be capable of localizing human singing, as both classes share semantic similarities in audio and visual features.

7. Conclusion

We revisit the problem of unsupervised visual sound source localization. For this task, we introduce a new large-scale benchmark called VGG-Sound Source, which is more challenging than existing ones such as Flickr SoundNet. We also suggest a simple, general and effective technique that significantly boosts the performance of existing sound source locators, by explicitly mining for hard negative image locations in the same image that contains the sounding objecs. A careful implementation of this idea using Tri-maps and differentiable thresholding allows us to significantly outperform the state of the art.

Acknowledgements

This work is supported by the UK EPSRC CDT in Autonomous Intelligent Machines and Systems, the Oxford-Google DeepMind Graduate Scholarship, the Google PhD Fellowship, and EPSRC Programme Grants Seebibyte EP/M013774/1 and VisualAI EP/T028572/1.

References

[1] Triantafyllos Afouras, Andrew Owens, Joon Son Chung, and Andrew Zisserman. Self-supervised learning of audio-visual objects from video. In *Proc. ECCV*, 2020. 1, 2, 3, 4, 7

[2] Relja Arandjelovic and Andrew Zisserman. Objects that sound. In *Proc. ECCV*, 2017. 1, 2, 3, 4, 7

[3] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *NeurIPS*, 2016. 2, 6

[4] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. VGG-Sound: A large-scale audio-visual dataset. In *Proc. ICASSP*, 2020. 2, 5

[5] Yung-Yu Chuang, Aseem Agarwala, Brian Curless, David H. Salesin, and Richard Szeliski. Video matting of complex scenes. *ACM Trans. Graph*, 2002. 5

[6] Joon Son Chung and Andrew Zisserman. Lip reading in the wild. In *Proc. ACCV*, 2016. 2

[7] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 2005. 2

[8] Abhishek Dutta and Andrew Zisserman. The via annotation software for images, audio and video. In *Proc. ACMM*, 2019. 5

[9] John W Fisher III, Trevor Darrell, William T Freeman, and Paul A Viola. Learning joint statistical models for audio-visual fusion and segregation. In *NeurIPS*, 2000. 2

[10] Chuang Gan, Hang Zhao, Peihao Chen, David Cox, and Antonio Torralba. Self-supervised moving vehicle tracking with stereo sound. In *Proc. ICCV*, 2019. 2

[11] Ruohan Gao and Kristen Grauman. 2.5d visual sound. In *Proc. CVPR*, 2019. 2

[12] J Gemmeke, D Ellis, D Freedman, A Jansen, W Lawrence, C Moore, M Plakal, and M Ritter. Audio Set: An ontology and human-labeled dataset for audio events. In *Proc. ICASSP*, 2017. 2

[13] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. CVPR*, 2014. 2

[14] David Harwath, Adria Recasens, Dídac Surís, Galen Chuang, Antonio Torralba, and James Glass. Jointly discovering visual objects and spoken words from raw sensory input. In *Proc. ECCV*, 2018. 1, 2, 4, 7

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 6

[16] John R. Hershey and Javier R. Movellan. Audio-vision: Locating sounds via audio-visual synchrony. In *NeurIPS*, 1999. 2

[17] Di Hu, Feiping Nie, and Xuelong Li. Deep multimodal clustering for unsupervised audiovisual learning. In *Proc. CVPR*, June 2019. 1, 2, 3, 6, 7

[18] Hamid Izadinia, Imran Saleemi, and Mubarak Shah. Multimodal analysis for identification and segmentation of moving-sounding objects. *IEEE Trans. Multimed.*, 2012. 2

[19] Naji Khosravan, Shervin Ardeshir, and Rohit Puri. On attention modules for audio-visual synchronization. In *Proc. CVPR Workshop*, 2019. 2

[20] Einat Kidron, Yoav Y Schechner, and Michael Elad. Pixels that sound. In *Proc. CVPR*, 2005. 2

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollr. Focal loss for dense object detection. In *Proc. ICCV*, 2017. 2

[22] Etienne Marcheret, Gerasimos Potamianos, Josef Vopicka, and Vaibhava Goel. Detecting audio-visual synchrony using deep neural networks. In *Proc. ICSA*, 2015. 2

[23] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3

[24] Andrew Owens and Alexei A. Efros. Audio-visual scene analysis with self-supervised multisensory features. In *Proc. ECCV*, 2018. 1

[25] Andrew Owens and Alexei A. Efros. Audio-visual scene analysis with self-supervised multisensory features. In *Proc. ECCV*, 2018. 2

[26] Rui Qian, Di Hu, Heinrich Dinkel, Mengyue Wu, Ning Xu, and Weiyao Lin. Multiple sound sources localization from coarse to fine. In *Proc. ECCV*, 2020. 2, 3, 6, 7

[27] Janani Ramaswamy and Sukhendu Das. See the sound, hear the pixels. In *Proc. WACV*, 2020. 2

[28] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2016. 2, 5

[29] Andrew Rouditchenko, Hang Zhao, Chuang Gan, Josh McDermott, and Antonio Torralba. Self-supervised audio-visual co-segmentation. In *Proc. ICASSP*, 2019. 2

[30] Arda Senocak, Tae-Hyun Oh, Junsik Kim, Ming-Hsuan Yang, and In So Kweon. Learning to localize sound source in visual scenes. In *Proc. CVPR*, 2018. 1, 2, 3, 4, 5, 6, 7

[31] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proc. CVPR*, 2016. 2

[32] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *Proc. CVPR*, 2018. 5

[33] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: the new data in multimedia research. *Commun. ACM*, 2016. 2

[34] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *Proc. ECCV*, 2018. 2, 3

[35] Paul Viola and Michael Jones. Robust real-time object detection. In *Proc. SCTV Workshop*, 2001. 2

[36] Dingzeyu Li Yapeng Tian and Chenliang Xu. Unified multisensory perception: Weakly-supervised audio-visual video parsing. In *Proc. ECCV*, 2020. 2, 3

[37] Hang Zhao, Chuang Gan, Wei-Chiu Ma, and Antonio Torralba. The sound of motions. In *Proc. ICCV*, 2019. 2

[38] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *Proc. ECCV*, 2018. 2