# Audio−Visual Segmentation

⋆Jinxing Zhou[1,2], ⋆Jianyuan Wang[2,3], Jiayi Zhang[2,4], Weixuan Sun[2,3], Jing Zhang[3], Stan Birchfield[5], Dan Guo[1], Lingpeng Kong[6,7], ✉Meng Wang[1], and ✉Yiran Zhong[2,7]

[1]Hefei University of Technology, [2]SenseTime Research, [3]Australian National University, [4]Beihang University, [5]NVIDIA, [6]The University of Hong Kong, [7]Shanghai Artificial Intelligence Laboratory
{eric.mengwang, zhongyiran}@gmail.com

**Abstract.** We propose to explore a new problem called audio-visual segmentation (AVS), in which the goal is to output a pixel-level map of the object(s) that produce sound at the time of the image frame. To facilitate this research, we construct the first audio-visual segmentation benchmark (AVSBench), providing pixel-wise annotations for the sounding objects in audible videos. Two settings are studied with this benchmark: 1) semi-supervised audio-visual segmentation with a single sound source and 2) fully-supervised audio-visual segmentation with multiple sound sources. To deal with the AVS problem, we propose a new method that uses a temporal pixel-wise audio-visual interaction module to inject audio semantics as guidance for the visual segmentation process. We also design a regularization loss to encourage the audio-visual mapping during training. Quantitative and qualitative experiments on the AVS-Bench compare our approach to several existing methods from related tasks, demonstrating that the proposed method is promising for building a bridge between the audio and pixel-wise visual semantics. Code is available at https://github.com/OpenNLPLab/AVSBench.

**Keywords:** Audio-visual segmentation, Benchmarking, AVSBench.

## 1 Introduction

A human can classify an object not only from its visual appearance but also from the sound it makes. For example, when we hear a dog bark or a siren wail, we know the sound is from a dog or ambulance, respectively. Such observations confirm that the audio and visual information complement each other.

To date, researchers have approached this problem of audio-visual classification from somewhat simplified scenarios. Some researchers have investigated audio-visual correspondence (AVC) [2,3,4] problem, which aims to determine whether an audio signal and a visual image describe the same scene. AVC is

⋆Equal contribution. ✉Corresponding author. This work is done when Jinxing Zhou is an intern at SenseTime Research.

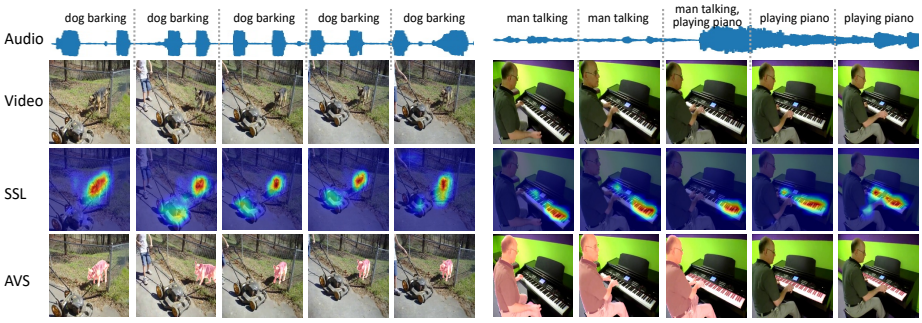


**Fig. 1. Comparison of the proposed AVS task with the SSL task.** Sound source localization (SSL) estimates a rough location of the sounding objects in the visual frame, at a patch level. We propose AVS to estimate pixel-wise segmentation masks for all the sounding objects, no matter the number of visible sounding objects.

based on the phenomenon that these two signals usually occur simultaneously, *e.g.*, a barking dog and a humming car. Others studied audio-visual event localization (AVEL) [20,22,38,42,44,45,30,31,9,51], which classifies the segments of a video into the pre-defined event labels. Similarly, some people have also explored audio-visual video parsing (AVVP) [37,41,21,46], whose goal is to divide a video into several events and classify them as audible, visible, or both. Due to a lack of pixel-level annotations, all these scenarios are restricted to the frame/temporal level, thus reducing the problem to that of audible image classification.

A related problem, known as sound source localization (SSL), aims to locate the visual regions within the frames that correspond to the sound [2,3,8,5,17,29]. Compared to AVC/AVEL/AVVP, the problem of SSL seeks patch-level scene understanding, *i.e.*, the results are usually presented by a heat map that is obtained either by visualizing the similarity matrix of the audio feature and the visual feature map, or by class activation mapping (CAM) [50]. It does not consider the actual shape of the sounding objects.

In this paper, we propose the pixel-level audio-visual segmentation (AVS) problem, which requires the network to densely predict whether each pixel corresponds to the given audio, so that a mask of the sounding object(s) is generated. Fig. 1 illustrates the differences between AVS and SSL. The AVS task is more challenging than previous tasks as it requires the network to not only locate the audible frames but also delineate the shape of the sounding objects.

To facilitate this research, we propose AVSBench, the first pixel-level audio-visual segmentation benchmark that provides ground truth labels for sounding objects. We divide our AVSBench dataset into two subsets, depending on the number of sounding objects in the video (single- or multi-source). With AVSBench, we study two settings of audio-visual segmentation: 1) semi-supervised Single Sound Source Segmentation (S4), and 2) fully-supervised Multiple Sound Source Segmentation (MS3). For both settings, the goal is to segment the object(s) from the visual frames that are producing sounds. We test six methods

from related tasks on AVSBench and provide a new AVS method as a strong baseline. The latter utilizes a standard encoder-decoder architecture but with a new temporal pixel-wise audio-visual interaction (TPAVI) module to better introduce the audio semantics for guiding visual segmentation. We also propose a loss function to utilize the correlation of audio-visual signals, which further enhances segmentation performance. We conduct extensive experiments on the AVSBench dataset to verify the benefits of considering audio signals for visual segmentation, and the effectiveness of our proposed approach.

## 2 Related Work

**Sound Source Localization (SSL).** The most related problem to ours is SSL, which aims to locate the regions in the visual frames that make sounds. Here we focus on SSL with multiple sources, which requires to accurately localize the true sound maker when there are multiple potential candidates [17,1,29,18]. As a pioneer, Hu *et al*. [17] divide the audio and visual features into multiple cluster centers and take the center distance as a supervision signal to rank the paired audio-visual information. Some methods adopt a two-stage strategy that first learn some prior knowledge of audio-visual correspondence from single sound source scenes, and then use that for multiple sound sources localization [18,29]. Rouditchenko *et al*. [33] tackle this problem by disentangling category concepts in the neural networks. This method is actually more related to the task of sound source separation [49,11,48,12] and shows sub-optimal performance regarding visual localization. It is worth noting that these SSL methods cannot clearly delineate the shape of the objects. Rather, the location map is computed by the audio-visual similarity matrix from a low resolution [2,3,36,8,28,5]. To overcome these limitations, this paper provides an audio-visual segmentation dataset with pixel-level ground truth to enable more accurate segmentation predictions.

**Audio-Visual Dataset.** To the best of our knowledge, there are no publicly available datasets that provide segmentation masks for the sounding visual objects with audio signals. Here we briefly introduce the popular datasets in the audio-visual community. For example, the AVE [38] and LLP [37] datasets are respectively collected for audio-visual event localization and video parsing tasks. They only have category annotations for video frames, and hence cannot be used for pixel-level segmentation. For sound source localization problem, researchers usually use the Flickr-SoundNet [36] and VGG-SS [5] datasets, where the videos are sampled from the large-scale Flickr [4] and VGGSound [6] datasets, respectively. The authors provide bounding boxes to outline the location of the target sound source, which could serve as patch-level supervision. However, this still inevitably suffers from incorrect evaluation results since the sounding objects are usually irregular in shape and some regions within the bounding box actually do not belong to the real sound source. This is a reason why current sound source localization methods can only roughly locate sounding objects but cannot learn their accurate shapes, which inhibits the mapping from audio signals to fine-grained visual cues.

**Table 1. AVSBench statistics**. The videos are split into train/valid/test. The asterisk (*) indicates that, for Single-source training, one annotation per video is provided; all others contain 5 annotations per video. (Since there are 5 clips per video, this is 1 annotation per clip.) Together, these yield the total annotated frames.

| subset | classes | videos | train/valid/test | annotated frames |
|---|---|---|---|---|
| Single-source | 23 | 4,932 | 3,452*/740/740 | 10,852 |
| Multi-sources | 23 | 424 | 296/64/64 | 2,120 |

**Table 2. Existing audio-visual dataset statistics.** Each benchmark is shown with the number of videos and the *annotated* frames. The final column indicates whether the frames are labeled by category, bounding boxes, or pixel-level masks.

| benchmark | videos | frames | classes | types | annotations |
|---|---|---|---|---|---|
| AVE [38] | 4,143 | 41,430 | 28 | video | category |
| LLP [37] | 11,849 | 11,849 | 25 | video | category |
| Flickr-SoundNet [36] | 5,000 | 5,000 | 50 | image | bbox |
| VGG-SS [5] | 5,158 | 5,158 | 220 | image | bbox |
| AVSBench (ours) | 5,356 | 12,972 | 23 | video | pixel |

## 3 The AVSBench

### 3.1 Dataset Statistics

AVSBench is designed for pixel-level audio-visual segmentation. We collected the videos using the techniques introduced in VGGSound [6] to ensure that the audio and visual clips correspond to the intended semantics. AVSBench contains two subsets—Single-source and Multi-sources—depending on the number of sounding objects. All videos were downloaded from YouTube with the *Creative Commons* license, and each video was trimmed to 5 seconds. The Single-source subset contains 4,932 videos over 23 categories, covering sounds from humans, animals, vehicles, and musical instruments. We provide the category names and the video number for each category in the supplemental material. For the Multi-sources subset, we picked the videos that contain multiple sounding objects, *e.g.*, a video of baby laughing, man speaking, and then woman singing. To be specific, we randomly chose two or three category names from the Single-source subset as keywords to search for online videos, then manually filtered out videos to ensure 1) each video has multiple sound sources, 2) the sounding objects are visible, and 3) there is no deceptive sound, *e.g.*, canned laughter. In total, this process yielded 424 videos for the Multi-sources subset, out of more than six thousand candidates. The ratio of train/validation/test split percentages are set as 70/15/15 for both subsets, as shown in Table 1. Several video examples are visualized in Fig. 2, where the red text indicates the name of sounding objects.

In addition, we make a comparison between AVSBench with other popular audio-visual benchmarks in Table 2. The AVE [38] dataset contains 4,143


**Fig. 2. AVSBench samples**. The AVSBench dataset contains the Single-source subset (Left) and Multi-sources subset (Right). Each video is divided into 5 clips, as shown. Annotated clips are indicated by brown framing rectangles; the name of sounding objects is indicated by red text. Note that for Single-source training set, only the first frame of each video is annotated, whereas 5 frames are annotated for all other sets.

videos covering 28 event categories. The LLP [37] dataset consists of 11,849 YouTube video clips spanning with 25 categories, collected from AudioSet [13]. Both the AVE and LLP datasets are labelled at a frame level, through audio-visual event boundaries. Meanwhile, the Flickr-SoundNet [36] dataset and VGG-SS [5] dataset are proposed for sound source localization (SSL), labelled at a patch level through bounding boxes. In contrast, our AVSBench contains 5,356 videos with 12,972 pixel-wise annotated frames. The benchmark is designed to facilitate the research on fine-grained audio-visual segmentation. Additionally, it provides accurate ground truth for SSL, which could help the training of SSL methods and serve as an evaluation benchmark for that problem as well.

### 3.2 Annotation

We divide each 5-second video into five equal 1-second clips, and we provide manual pixel-level annotations for the last frame of each clip. For this sampled frame, the ground truth label is a binary mask indicating the pixels of sounding objects, according to the audio at the corresponding time. For example, in the Multi-sources subset, even though a dancing person shows drastic movement spatially, it would not be labelled as long as no sound was made. In clips where objects do not make sound, the object should not be masked, *e.g.*, the *piano* in the first two clips of the last row of Fig. 2b. Similarly, when more than one object emits sound, all the emitting objects are annotated, *e.g.*, the guitar and ukulele in the first row in Fig. 2b. Also, when the sounding objects in the video are dynamically changing, the difficulty is further increased, *e.g.*, the second, third,

and fourth rows in Fig. 2b. Currently, for large-scale objects, we only annotate their most representative parts. For example, we label the keyboard of pianos because it is sufficiently recognizable, while the cabinet part is often too varied.

For the videos in the training split of Single-source, we only annotate the first frame (with the assumption that the information from one-shot annotation is sufficient, as the Single-source subset has a single and consistent sounding object over time). This assumption is verified by the quantitative experimental results shown in Table 3. For the more challenging Multi-sources subset, all clips are annotated for training, since the sounding objects may change over time. Note that for validation and test splits, all clips are annotated, as shown in Table 1.

### 3.3 Two Benchmark Settings

We provide two benchmark settings for our AVSBench: the semi-supervised Single Sound Source Segmentation (S4) and the fully supervised Multiple Sound Source Segmentation (MS3). For ease of expression, we denote the video sequence as $S$, which consists of $T$ non-overlapping yet continuous clips $\{S_t^v, S_t^a\}_{t=1}^T$, where $S^v$ and $S^a$ are the visual and audio components, and $T = 5$. In practice, we extract the video frame at the end of each second. **Semi-supervised S4** corresponds to the Single-source subset. It is termed as semi-supervised because only part of the ground truth is given during training (*i.e.*, the first sampled frame of the videos) but all the video frames require a prediction during evaluation. We denote the pixel-wise label as $\boldsymbol{Y}_{t=1}^s \in \mathbb{R}^{H \times W}$, where $H$ and $W$ are the frame height and width, respectively. $\boldsymbol{Y}_{t=1}^s$ is a binary matrix where 1 indicates sounding objects while 0 corresponds to background or silent objects. **Fully-supervised MS3** deals with the Multi-sources subset, where the labels of all five frames of each video are available for training. The ground truth is denoted as $\{\boldsymbol{Y}_t^m\}_{t=1}^T$, where $\boldsymbol{Y}_t^m \in \mathbb{R}^{H \times W}$ is the binary label for the $t$-th video clip.

The goal for both settings is to correctly segment the sounding object(s) for each video clip with the audio and visual cues, *i.e.*, $S^a$ and $S^v$. Generally, $S^a$ is expected to indicate the target object, while $S^v$ provides information for fine-grained segmentation. The predictions are denoted as $\{\boldsymbol{M}_t\}_{t=1}^T$, $\boldsymbol{M}_t \in \mathbb{R}^{H \times W}$.

## 4 A Baseline

We propose a new baseline method for the pixel-level audio-visual segmentation (AVS) task as shown in Fig. 3. We use the same framework in both semi- and fully-supervised settings. Following the convention of semantic segmentation methods [24,32,39,43], our method adopts an encoder−decoder architecture.

**The Encoder:** We extract audio and visual features independently. Given an audio clip $S^a$, we first process it to a spectrogram via the short-time Fourier transform, and then send it to a convolutional neural network, VGGish [16]. We use the weights that are pretrained on AudioSet [13] to extract audio features $\boldsymbol{A} \in \mathbb{R}^{T \times d}$, where $d = 128$ is the feature dimension. For a video frame $S^v$, we

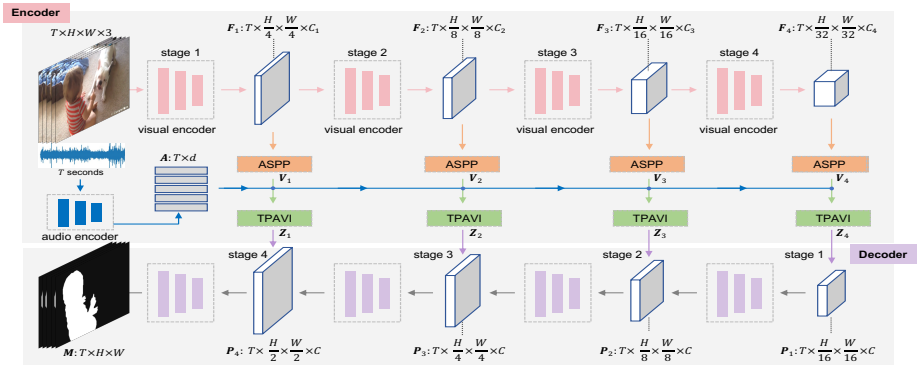


**Fig. 3. Overview of the Baseline**, which follows a hierarchical Encoder-Decoder pipeline. The *encoder* takes the video frames and the entire audio clip as inputs, and outputs visual and audio features, respectively denoted as $\boldsymbol{F}_i$ and $\boldsymbol{A}$. The visual feature map $\boldsymbol{F}_i$ at each stage is further sent to the ASPP [7] module and then our TPAVI module (introduced in Sec. 4). ASPP provides different receptive fields for recognizing visual objects, while TPAVI focuses on the temporal pixel-wise audio-visual interaction. The *decoder* progressively enlarges the fused feature maps by four stages and finally generates the output mask $\boldsymbol{M}$ for sounding objects.

extract visual features with popular convolution-based or vision transformer-based backbones. We try both two options in the experiments and they show similar performance trends. These backbones produce hierarchical visual feature maps during the encoding process, as shown in Fig. 3. We denote the features as $\boldsymbol{F}_i \in \mathbb{R}^{T\times h_i\times w_i\times C_i}$, where $(h_i, w_i) = (H, W)/2^{i+1}$, $i = 1, \ldots, n$. The number of levels is set to $n = 4$ in all experiments.

**Cross-Modal Fusion:** We use Atrous Spatial Pyramid Pooling (ASPP) modules [7] to further post-process the visual features $\boldsymbol{F}_i$ to $\boldsymbol{V}_i \in \mathbb{R}^{T\times h_i\times w_i\times C}$, where $C = 256$. These modules employ multiple parallel filters with different rates and hence help to recognize visual objects with different receptive fields, *e.g.*, different sized moving objects.

Then, we consider introducing the audio information to build the audio-visual mapping to assist with identifying the sounding object. This is particularly essential for the MS3 setting where there are multiple dynamic sound sources. Our intuition is that, although the auditory and visual signals of the sound sources may not appear simultaneously, they usually exist in more than one video frame. Therefore, integrating the audio and visual signals of the whole video should be beneficial. Motivated by [40] that uses the non-local block to encode space-time relation, we adopt a similar module to encode the temporal pixel-wise audio-visual interaction (TPAVI). The current visual feature map $\boldsymbol{V}_i$ and the audio feature $\boldsymbol{A}$ of the entire video are sent into the TPAVI module. Specifically, the audio feature $\boldsymbol{A}$ is first transformed to a feature space with the same dimension as the visual feature $\boldsymbol{V}_i$, by a linear layer. Then it is spatially duplicated $h_i w_i$ times and reshaped to the same size as $\boldsymbol{V}_i$. We denote such

processed audio feature as $\hat{\boldsymbol{A}}$. Next, it is expected to find those pixels of visual feature map $\boldsymbol{V}_i$ that have a high response to the audio counterpart $\hat{\boldsymbol{A}}$ through the entire video.

Such an audio-visual interaction can be measured by dot-product, then the updated feature maps $\boldsymbol{Z}_i$ at the $i$-th stage can be computed as,

$$\boldsymbol{Z}_i = \boldsymbol{V}_i + \mu(\alpha_i \; g(\boldsymbol{V}_i)), \text{ where } \alpha_i = \frac{\theta(\boldsymbol{V}_i) \; \phi(\hat{\boldsymbol{A}})^\top}{N} \tag{1}$$

where $\theta$, $\phi$, $g$, and $\mu$ are $1 \times 1 \times 1$ convolutions, $N = T \times h_i \times w_i$ is a normalization factor, $\alpha_i$ denotes the audio-visual similarity, and $\boldsymbol{Z}_i \in \mathbb{R}^{T \times h_i \times w_i \times C}$. Each visual pixel interacts with all the audios through the TPAVI module. We provide a visualization of the audio-visual attention in TPAVI later in Fig. 6, which shows a similar "appearance" to the prediction of SSL methods because it constructs a pixel to audio mapping.

**The Decoder:** We adopt the decoder of Panoptic-FPN [19] in this work for its flexibility and effectiveness, though any valid decoder architecture could be used. In short, at the $j$-th stage, where $j = 2, 3, 4$, both the outputs from stage $\boldsymbol{Z}_{5-j}$ and the last stage $\boldsymbol{Z}_{6-j}$ of the encoder are utilized for the decoding process. The decoded features are then upsampled to the next stage. The final output of the decoder is $\boldsymbol{M} \in \mathbb{R}^{T \times H \times W}$, activated by *sigmoid*.

**Objective function:** Given the prediction $\boldsymbol{M}$ and the pixel-wise label $\boldsymbol{Y}$, we adapt the binary cross entropy (BCE) loss as the main supervision function. Besides, we use an additional regularization term $\mathcal{L}_{\text{AVM}}$ to force the audio-visual mapping. Specifically, we use the Kullback–Leibler (KL) divergence to ensure the masked visual features have similar distributions with the corresponding audio features. In other words, if the audio features of some frames are close in feature space, the corresponding sounding objects are expected to be close in feature space. The total objective function $\mathcal{L}$ can be computed as follows:

$$\mathcal{L} = \text{BCE}(\boldsymbol{M}, \boldsymbol{Y}) + \lambda \mathcal{L}_{\text{AVM}}(\boldsymbol{M}, \boldsymbol{Z}, \boldsymbol{A}), \tag{2}$$

$$\mathcal{L}_{\text{AVM}} = \text{KL}(avg \; (\sum_{i=1}^{n} \boldsymbol{M}_i \odot \boldsymbol{Z}_i), \boldsymbol{A}_i), \tag{3}$$

where $\lambda$ is a balance weight, $\odot$ denotes element-wise multiplication, and *avg* denotes the average pooling operation. At each stage, we down-sample the prediction $\boldsymbol{M}$ to $\boldsymbol{M}_i$ via average pooling to have the same shape as $\boldsymbol{Z}_i$. The vector $\boldsymbol{A}_i$ is a linear transformation of $\boldsymbol{A}$ that has the same feature dimension with $\boldsymbol{Z}_i$. For the semi-supervised S4 setting, we found that the audio-visual regularization loss does not help, so we set $\lambda = 0$ in this setting.

## 5 Experimental Results

### 5.1 Implementation details

We conduct training and evaluation on the proposed AVSBench dataset, with both convolution-based and transformer-based backbones, ResNet-50 [15] and

**Table 3. Comparison with methods from related tasks.** Results of mIoU (%) and F-score under both S4 and MS3 settings are reported.

| Metric | Setting | SSL | | VOS | | SOD | | AVS (ours) | |
|---|---|---|---|---|---|---|---|---|---|
| | | LVS[5] | MSSL[29] | 3DC[26] | SST[10] | iGAN[27] | LGVT[47] | ResNet50 | PVT-v2 |
| mIoU | S4 | 37.94 | 44.89 | 57.10 | 66.29 | 61.59 | 74.94 | 72.79 | **78.74** |
| | MS3 | 29.45 | 26.13 | 36.92 | 42.57 | 42.89 | 40.71 | 47.88 | **54.00** |
| F-score | S4 | .510 | .663 | .759 | .801 | .778 | .873 | .848 | **.879** |
| | MS3 | .330 | .363 | .503 | .572 | .544 | .593 | .578 | **.645** |

Pyramid Vision Transformer (PVT-v2) [39]. The backbones have been pretrained on the ImageNet [34] dataset. All the video frames are resized to the shape of $224 \times 224$. The channel sizes of the four stages are $C_{1:4} = [256, 512, 1024, 2048]$ and $C_{1:4} = [64, 128, 320, 512]$ for ResNet-50 and PVT-v2, respectively. The channel size of the ASPP module is set to $C = 256$. We use the VGGish model to extract audio features, a VGG-like network [16] pretrained on the AudioSet [13] dataset. The audio signals are converted to one-second splits as the network inputs. We use the Adam optimizer with a learning rate of 1e-4 for training. The batch size is set to 4 and the number of training epochs are 15 and 30 respectively for the semi-supervised S4 and the fully-supervised MS3 settings. The $\lambda$ in Eq. (2) is empirically set to 0.5.

### 5.2 Comparison with methods from related tasks

We compare our baseline framework with the methods from three related tasks, including sound source localization (SSL), video object segmentation (VOS), and salient object detection (SOD). For each task, we report the results of two methods on our AVSBench benchmark, *i.e.*, LVS [5] and MSSL [29] for SSL, 3DC [26] and SST [10] for VOS, iGAN [27] and LGVT [47] for SOD. We select these methods as they are the SOTA in their fields: 1) *LVS* uses the background and the most confident regions of sounding objects to design a contrastive loss for audio-visual representation learning. The localization map is obtained by computing the audio-visual similarity. 2) *MSSL* is a two-stage method for multiple sound source localization and the localization map is obtained by Grad-CAM [35]. 3) *3DC* adopts an architecture that is fully constructed by powerful 3D convolutions to encode video frames and predict segmentation masks. 4) *SST* introduces a transformer architecture to achieve sparse attention of the features in the spatiotemporal domain. 5) *iGAN* is a ResNet-based generative model for saliency detection, considering the inherent uncertainty of saliency detection. 6) *LGVT* is a saliency detection method based on Swin transformer [23], whose long-range dependency modeling ability leads to a better global context modeling. We adopt the architecture of these methods and fit them to our semi-supervised S4 and fully-supervised MS3 settings. For a fair comparison, the backbones of these methods are all pretrained on the ImageNet [34].

**Table 4. Impact of audio signal and TPAVI.** Results (mIoU) of AVS with and without the TPAVI module. The middle row indicates directly adding the audio and visual features, which already improves performance under the MS3 setting. The TPAVI module further enhances the results over all settings and backbones.

| AVS method | S4 | | MS3 | |
|---|---|---|---|---|
| | ResNet50 | PVT-v2 | ResNet50 | PVT-v2 |
| without TPAVI | 70.12 | 77.76 | 43.56 | 48.21 |
| with A⊕V | 70.54 | 77.65 | 45.69 | 51.55 |
| with TPAVI | **72.79** | **78.74** | **46.64** | **53.06** |

The quantitative results are shown in Table 3, with Mean Intersection over Union (mIoU) and F-score[1] as the evaluation metrics. There is a substantial gap between the results of SSL methods and those of our baseline, mainly because the SSL methods cannot provide pixel-level prediction. Also, our baseline framework shows a consistent superiority to the VOS and SOD methods in both settings. It is worth noting that the state-of-the-art SOD method LGVT [47] slightly outperforms our ResNet50-based baseline on the Single-source set (74.94% mIoU *vs.* 72.79% mIoU), mainly because LGVT uses the strong Swin Transformer backbone [23]. However, when it comes to the Multi-sources setting, the performance of LGVT is obviously worse than that of our ResNet50-based baseline (40.71% mIoU *vs.* 47.88% mIoU). This is because the SOD method relies on the dataset prior, and cannot handle the situations where sounding objects change but visual contents remain the same (further discussed in the supplementary material). Instead, the audio signals guide our method to identify which object to segment, leading to better performance. Moreover, if also using a transformer-based backbone, our method is stronger than LGVT in both settings. Besides, we notice that although SSL methods utilize both the audio and visual signals, they cannot match the performance of VOS or SOD methods that only use visual frames. It indicates the significance of pixel-wise scene understanding. The proposed AVS baselines achieve satisfactory performance under the semi-supervised S4 setting (around 70% mIoU), which verifies that one-shot annotation is sufficient for single-source cases. Some qualitative examples are provided in our supplementary material to compare the proposed baseline with these methods.

### 5.3 Ablation Study

**Impact of audio signal and TPAVI**. As introduced in Sec. 4, the TPAVI module is designed to formulate the audio-visual interactions from a temporal and pixel-wise level, introducing the audio information to explore the visual segmentation. We verify its impact in Table 4. Two rows show the proposed AVS method with or without the TPAVI module, while "A⊕V" indicates directly

[1] F-score considers both the precision and recall: $F_\beta = \frac{(1+\beta^2)\times \text{precision} \times \text{recall}}{\beta^2 \times \text{precision} + \text{recall}}$, where $\beta^2$ is set to 0.3 in our experiments.

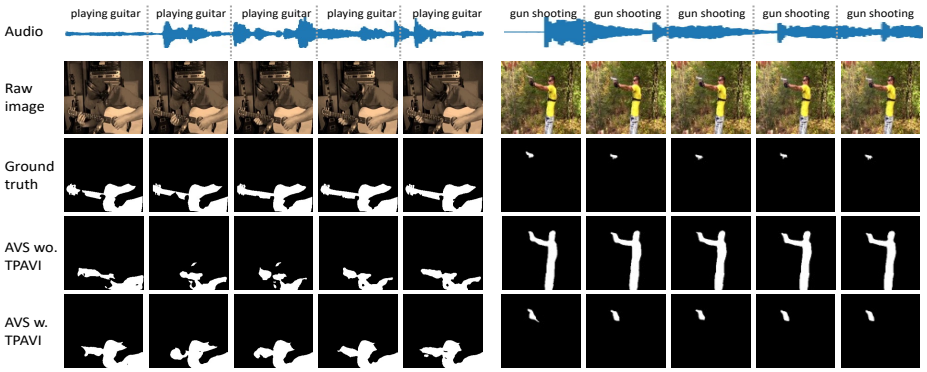


**Fig. 4. Qualitative results under the semi-supervised S4 setting**. Two benefits are noticed by introducing the audio signal (TPAVI): 1) learning the shape of the sounding object, *e.g.*, *guitar* in the video (LEFT); 2) segmenting according to the correct sound source, *e.g.*, the *gun* rather than the *man* (RIGHT).

adding the audio to visual features. It will be noticed that adding the audio features to the visual ones does not result in a clear difference under the S4 setting, but lead to a distinct gain under the MS3 setting. This is consistent with our hypothesis that audio is especially beneficial to samples with multiple sound sources, because the audio signals can guide which object(s) to segment. Furthermore, with TPAVI, each visual pixel hears the current sound and the sounds at other times, while simultaneously interacting with other pixels. The pixels with high similarity to the same sound are more likely to belong to one object. TPAVI helps further enhance the performance over various settings and backbones, *e.g.*, 72.79% *vs.* 70.54% when using ResNet50 as the backbone under the S4 setting, and 53.06% *vs.* 51.55% if using PVT-v2 under the MS3 setting. Additionally, it is worth noting that the convolution blocks in the TPAVI module allow to project the input visual and audio features to the latent spaces that are suitable for attention computation. For instance, under the S4 setting and using ResNet50, if abandoning the four convolution blocks ($\theta$, $\phi$, $g$, and $\mu$) in the TPAVI module, the mIoU will significantly drop from 72.79% to 59.21%.

We also visualize some qualitative examples to reflect the impact of TPAVI. As shown in Fig. 4, the AVS method with TPAVI depicts the shape of sounding object better, *e.g.*, the *guitar* in the left video, while it can only segment several parts of the guitar without TPAVI. Such benefit can also be observed in MS3 setting, as shown in Fig. 5, the model enables to ignore those pixels of *human hands* with TPAVI. More importantly, with TPAVI, the model is able to segment the correct sounding object and ignore the potential sound sources which actually do not make sounds, *e.g.*, the *man* on the right of Fig. 4. Also, the "AVS w. TPAVI" has stronger ability to capture multiple sound sources. As shown on the right of Fig. 5, the *person* who is singing is mainly segmented with TPAVI but is almost lost without TPAVI.

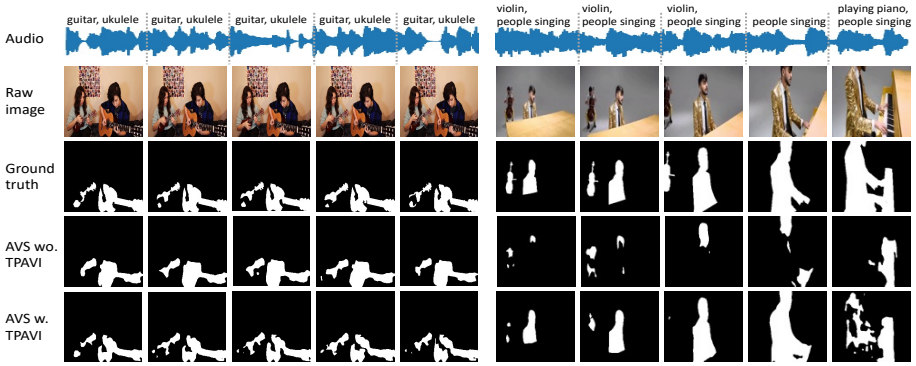


**Fig. 5. Qualitative results under the fully-supervised MS3 setting**. Note that AVS with TPAVI uses audio information to perform better in terms of 1) filtering out the distracting visual pixels that do not correspond to the audio, *i.e.*, the *human hands* (Left); 2) segmenting the correct sound source in the visual frames that matches the audio more accurately, *i.e.*, the *singing person* (Right).

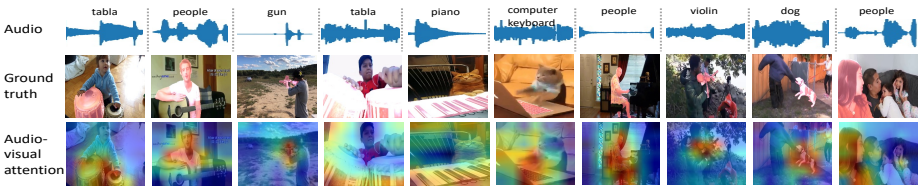


**Fig. 6. Audio-Visual attention maps from the fourth stage TPAVI.** A brighter color indicates a higher response. Such heatmaps are adopted as the final results for the SSL task, but just the intermediate output of the TPAVI module. TPAVI helps the model focus more on the visual regions that are semantic-corresponding to the audio.

Besides, we also visualize the audio-visual attention matrices to explore what happens in the cross-modal fusion process of TPAVI. As shown in Fig. 6, the high response area basically overlaps the region of sounding objects. The attention matrix is obtained from $\alpha_i$ in Eq. (1) of the fourth stage TPAVI. This is visually similar to the localization heatmap of these SSL methods, but only the intermediate result in our AVS method. It suggests that TPAVI builds a mapping from the visual pixels to the audio signals, which is semantically consistent.

**Comparison with a two-stage baseline.** The AVS task can also be tackled by two stages. In the first stage, an off-the-shelf segmentation model extracts instance segmentation maps. Then, the maps and visual features from the first stage are concatenated with audios, and fed into a PVT-v2 structure to predict the final results. We denote this method as TwoSep, and the results are shown in Table 5. It indicates the AVS task is *Not* bottlenecked by the segmentation quality, as the final performance is almost unchanged with a much stronger Mask R-CNN (backbone from ResNet50 to powerful ResNeXt101), *e.g.*, mIoU 50.32%

**Table 5. Comparison with a two-stage baseline (TwoSep)**, which first generates instance segmentation maps by off-the-shelf Mask R-CNN [14] and then combines the audio signal for final sounding objects segmentation. Its performance is not bottlenecked by the segmentation quality (using different Mask-RCNN backbones) but is largely influenced by the audio signal.

| Metric | Setting | TwoSep wo. audio | | TwoSep w. audio | | Ours |
|---|---|---|---|---|---|---|
| | | Res50 | ResNeXt101 | Res50 | ResNeXt101 | - |
| mIoU | S4 | 69.56 | 69.98 | 71.73 | 71.81 | **78.74** |
| | MS3 | 47.25 | 47.40 | 50.32 | 50.22 | **54.00** |

**Table 6. Effectiveness of $\mathcal{L}_{\mathbf{AVM}}$.** The two variants of $\mathcal{L}_{\text{AVM}}$ both bring a clear performance gain compared with only using a standard BCE loss.

| Objective function | MS3 (mIoU) | | MS3 (F-score) | |
|---|---|---|---|---|
| | ResNet50 | PVT-v2 | ResNet50 | PVT-v2 |
| $\mathcal{L}_{\text{BCE}}$ | 46.64 | 53.06 | .558 | .626 |
| $\mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{AVM-VV}}$ | 46.71 | 53.77 | .577 | .644 |
| $\mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{AVM-AV}}$ | **47.88** | **54.00** | **.578** | **.645** |

*vs.* 50.22% in MS3 setting. Instead, without or with audios would largely affect the performance, *e.g.*, mIoU 47.25% *vs.* 50.32%. This again reflects the positive impact of audio signals, especially in the MS3 setting.

**Effectiveness of $\mathcal{L}_{\mathbf{AVM}}$.** We expect that constructing the mapping between audio and visual features will enhance the network's ability to identify the correct objects. Therefore, we propose a $\mathcal{L}_{\text{AVM}}$ loss to introduce a soft constraint. We only apply $\mathcal{L}_{\text{AVM}}$ in the fully-supervised MS3 setting because the change of sounding objects only happens there. As shown in Table 6, we explore two variants of the $\mathcal{L}_{\text{AVM}}$ loss. $\mathcal{L}_{\text{AVM-AV}}$ is the one introduced in Eq. (3). It encourages the visual features masked by the segmentation result to be consistent with the corresponding audio features in a statistical way. Alternatively, $\mathcal{L}_{\text{AVM-VV}}$ first finds the closest audio partner for each candidate audio, and then computes the KL distance of the corresponding visual features (also masked by segmentation results). This is based on the idea that if two clips share similar audio signals, the visual features of their sounding objects should also be similar. As shown in Table 6, both variants achieve a clear performance gain. For example, $\mathcal{L}_{\text{AVM-AV}}$ improves the mIoU by around 1% and F-score by about 2%. In practice, we just use $\mathcal{L}_{\text{AVM-AV}}$, since $\mathcal{L}_{\text{AVM-VV}}$ inconveniently requires a ranking operation.

**Without backbone pre-training.** We try to train the AVS framework without the pretrained backbones. As expected, we observe an obvious performance drop, *e.g.*, the mIoU decreases from 72.79% to 44.05% under S4 setting with ResNet50 as backbone. We speculate that it is difficult for the model to learn the audio and visual representation totally from scratch, especially for this challenging pixel-wise segmentation task.

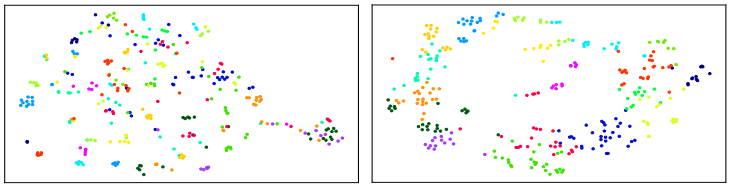


**Fig. 7. T-SNE [25] visualization of the visual features, trained with or without audio.** We divided the audio features into $K = 20$ clusters via PCA, and then assign the cluster labels to the corresponding visual features. The points with the same color share the same audio cluster labels. When training is accompanied by audio signals (right), the visual features illustrate a closer trend with the audio feature distribution, *i.e.*, points with the same colors gather together (Best viewed in color.)

**T-SNE visualization analysis.** On the test split of the Multi-sources set, we use the PVT-v2 based AVS model to obtain the visual features. Since the Multi-source set do not have category labels (its videos may contain several categories), we use the principal component analysis (PCA) to divide the audio features into $K = 20$ clusters. Then we assign the audio cluster labels to the corresponding visual features. In this case, if the audio and the visual features are correlated, the visual features should be clustered as well. We use the t-SNE visualization to verify this assumption. As shown in Fig. 7a, without audio signals, the learned visual features distribute chaotically; whereas in Fig. 7b, the visual features sharing the same audio labels tend to gather together.

## 6 Conclusion

We have proposed a new task called AVS, which aims to generate pixel-level binary segmentation masks for sounding objects in audible videos. To facilitate research in this area, we collected the first audio-visual segmentation benchmark (called AVSBench). We presented a new pixel-level AVS method to serve as a strong baseline, which includes a TPAVI module to encode the pixel-wise audio-visual interactions within temporal video sequences and a regularization loss to help the model learn audio-visual correlations. We compared our method with several existing state-of-the-art methods of the related tasks on AVSBench, and further demonstrated that our method can build a connection between the sound and the appearance of an object. For future work, we believe this research will pave the way for multimodal semantic segmentation.

**Acknowledgement** The research of Jinxing Zhou, Dan Guo, and Meng Wang was supported by the National Key Research and Development Program of China (2018YFB0804205), and the National Natural Science Foundation of China (72188101, 61725203). Thanks to the SenseTime Research for providing access to the GPUs used for conducting experiments. The article solely reflects the opinions and conclusions of its authors but not the funding agents.

## References


1. Afouras, T., Owens, A., Chung, J.S., Zisserman, A.: Self-supervised learning of audio-visual objects from video. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 208–224 (2020)
2. Arandjelovic, R., Zisserman, A.: Look, listen and learn. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 609–617 (2017)
3. Arandjelovic, R., Zisserman, A.: Objects that sound. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 435–451 (2018)
4. Aytar, Y., Vondrick, C., Torralba, A.: Soundnet: Learning sound representations from unlabeled video. Advances in Neural Information Processing Systems **29** (2016)
5. Chen, H., Xie, W., Afouras, T., Nagrani, A., Vedaldi, A., Zisserman, A.: Localizing visual sounds the hard way. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16867–16876 (2021)
6. Chen, H., Xie, W., Vedaldi, A., Zisserman, A.: VGGSound: A large-scale audio-visual dataset. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 721–725 (2020)
7. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(4), 834–848 (2017)
8. Cheng, Y., Wang, R., Pan, Z., Feng, R., Zhang, Y.: Look, listen, and attend: Co-attention network for self-supervised audio-visual representation learning. In: Proceedings of the 28th ACM International Conference on Multimedia (ACM). pp. 3884–3892 (2020)
9. Duan, B., Tang, H., Wang, W., Zong, Z., Yang, G., Yan, Y.: Audio-visual event localization via recursive fusion by joint co-attention. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 4013–4022 (2021)
10. Duke, B., Ahmed, A., Wolf, C., Aarabi, P., Taylor, G.W.: SSTVOS: Sparse spatiotemporal transformers for video object segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5912–5921 (2021)
11. Gao, R., Feris, R., Grauman, K.: Learning to separate object sounds by watching unlabeled video. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 35–53 (2018)
12. Gao, R., Grauman, K.: Co-separating sounds of visual objects. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3879–3888 (2019)
13. Gemmeke, J.F., Ellis, D.P., Freedman, D., Jansen, A., Lawrence, W., Moore, R.C., Plakal, M., Ritter, M.: Audio set: An ontology and human-labeled dataset for audio events. In: IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 776–780. IEEE (2017)
14. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)

16. Hershey, S., Chaudhuri, S., Ellis, D.P., Gemmeke, J.F., Jansen, A., Moore, R.C., Plakal, M., Platt, D., Saurous, R.A., Seybold, B., et al.: CNN architectures for large-scale audio classification. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 131–135 (2017)
17. Hu, D., Nie, F., Li, X.: Deep multimodal clustering for unsupervised audiovisual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9248–9257 (2019)
18. Hu, D., Qian, R., Jiang, M., Tan, X., Wen, S., Ding, E., Lin, W., Dou, D.: Discriminative sounding objects localization via self-supervised audiovisual matching. Advances in Neural Information Processing Systems (NeurIPS) **33**, 10077–10087 (2020)
19. Kirillov, A., Girshick, R., He, K., Dollár, P.: Panoptic feature pyramid networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6399–6408 (2019)
20. Lin, Y.B., Li, Y.J., Wang, Y.C.F.: Dual-modality seq2seq network for audio-visual event localization. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2002–2006. IEEE (2019)
21. Lin, Y.B., Tseng, H.Y., Lee, H.Y., Lin, Y.Y., Yang, M.H.: Exploring cross-video and cross-modality signals for weakly-supervised audio-visual video parsing. Advances in Neural Information Processing Systems **34** (2021)
22. Lin, Y.B., Wang, Y.C.F.: Audiovisual transformer with instance attention for audio-visual event localization. In: Proceedings of the Asian Conference on Computer Vision (ACCV) (2020)
23. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021)
24. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. CoRR **abs/1411.4038** (2014), `http://arxiv.org/abs/1411.4038`
25. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
26. Mahadevan, S., Athar, A., Ošep, A., Hennen, S., Leal-Taixé, L., Leibe, B.: Making a case for 3D convolutions for object segmentation in videos. arXiv preprint arXiv:2008.11516 (2020)
27. Mao, Y., Zhang, J., Wan, Z., Dai, Y., Li, A., Lv, Y., Tian, X., Fan, D.P., Barnes, N.: Transformer transforms salient object detection and camouflaged object detection. arXiv preprint arXiv:2104.10127 (2021)
28. Owens, A., Efros, A.A.: Audio-visual scene analysis with self-supervised multisensory features. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 631–648 (2018)
29. Qian, R., Hu, D., Dinkel, H., Wu, M., Xu, N., Lin, W.: Multiple sound sources localization from coarse to fine. In: Proceedings of the European conference on computer vision (ECCV). pp. 292–308. Springer (2020)
30. Ramaswamy, J.: What makes the sound?: A dual-modality interacting network for audio-visual event localization. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 4372–4376. IEEE (2020)
31. Ramaswamy, J., Das, S.: See the sound, hear the pixels. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 2970–2979 (2020)
32. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015)

33. Rouditchenko, A., Zhao, H., Gan, C., McDermott, J., Torralba, A.: Self-supervised audio-visual co-segmentation. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2357–2361. IEEE (2019)
34. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: ImageNet large scale visual recognition challenge. International Journal of Computer Vision **115**(3), 211–252 (2015)
35. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 618–626 (2017)
36. Senocak, A., Oh, T.H., Kim, J., Yang, M.H., Kweon, I.S.: Learning to localize sound source in visual scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4358–4366 (2018)
37. Tian, Y., Li, D., Xu, C.: Unified multisensory perception: Weakly-supervised audio-visual video parsing. In: Proceedings of the European conference on computer vision (ECCV). pp. 436–454. Springer (2020)
38. Tian, Y., Shi, J., Li, B., Duan, Z., Xu, C.: Audio-visual event localization in unconstrained videos. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 247–263 (2018)
39. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: PVTv2: Improved baselines with pyramid vision transformer. Computational Visual Media **8**(3), 1–10 (2022)
40. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7794–7803 (2018)
41. Wu, Y., Yang, Y.: Exploring heterogeneous clues for weakly-supervised audio-visual video parsing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1326–1335 (2021)
42. Wu, Y., Zhu, L., Yan, Y., Yang, Y.: Dual attention matching for audio-visual event localization. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 6292–6300 (2019)
43. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. arXiv preprint arXiv:2105.15203 (2021)
44. Xu, H., Zeng, R., Wu, Q., Tan, M., Gan, C.: Cross-modal relation-aware networks for audio-visual event localization. In: Proceedings of the 28th ACM International Conference on Multimedia (ACM). pp. 3893–3901 (2020)
45. Xuan, H., Zhang, Z., Chen, S., Yang, J., Yan, Y.: Cross-modal attention network for temporal inconsistent audio-visual event localization. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). pp. 279–286 (2020)
46. Yu, J., Cheng, Y., Zhao, R.W., Feng, R., Zhang, Y.: MM-Pyramid: Multimodal pyramid attentional network for audio-visual event localization and video parsing. arXiv preprint arXiv:2111.12374 (2021)
47. Zhang, J., Xie, J., Barnes, N., Li, P.: Learning generative vision transformer with energy-based latent space for saliency prediction. Advances in Neural Information Processing Systems (NeurIPS) **34** (2021)
48. Zhao, H., Gan, C., Ma, W.C., Torralba, A.: The sound of motions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1735–1744 (2019)

49. Zhao, H., Gan, C., Rouditchenko, A., Vondrick, C., McDermott, J., Torralba, A.: The sound of pixels. In: Proceedings of the European conference on computer vision (ECCV). pp. 570–586 (2018)
50. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2921–2929 (2016)
51. Zhou, J., Zheng, L., Zhong, Y., Hao, S., Wang, M.: Positive sample propagation along the audio-visual event line. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8436–8444 (2021)