Benchmarking Audio Visual Segmentation for Long-Untrimmed Videos

Supplementary Material

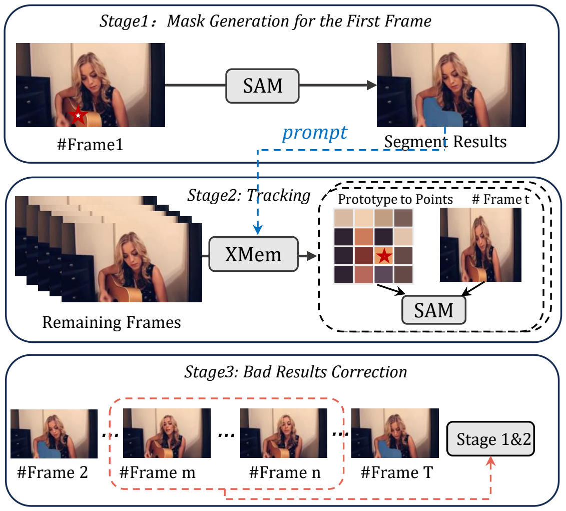

Figure 8. The flowchart of mask annotation tool. In the first stage, we manually annotate points to acquire the masks of audible objects in the first frame of audible segments. In stage two, these masks serve as prompts for XMem, which identifies the closest prototype for each subsequent frame. These prototypes are then transformed into points and combined with their frames in SAM to create high-quality masks. In the third stage, we identify and re-annotate low-quality segments in the video.

7. Mask Annotation Tool

Meticulous polygon-based mask annotation is a time-intensive and laborious process. Inspired by [49], we develop a semi-automatic annotation tool based on SAM [25] and the tacking method XMem [9]. This tool simplifies the annotation process, depicted in Figure 8.

In the first stage, we leverage SAM to segment the audible regions with the designated prompts, *i.e.* positive points and negative points. We develop a user interface to facilitate easy mask annotations through minimal interaction, as shown in Figure 9 (a). Here, masks for only the initial frame of each sound segment are produced to serve as preliminary references for the auditory objects. Subsequently, we employ XMem to extend these initial masks to the remaining frames. However, given that XMem is a semi-supervised method, the resultant tracking masks often suffer from poor quality. To counter this, we transform the XMem's prototype outputs into points, which we then use as image prompts to enhance the mask quality with SAM. In some cases, the annotated objects feature significant deformation, leading to tracking difficulties and substandard mask quality, as indicated in Figure 5. When such tracking failures occur, our correction interface (shown in Figure 9 (b)) enables annotators to pinpoint and re-annotate the failed clips.

8. More Details about the Strong Baseline

8.1. Architecture Details

Visual Branch. To obtain the masks and bounding boxes of potential sounding objects, we employ MaskFormer [7] and DETR [3] as the visual branches. Unlike typical visual datasets where all objects are labeled, in AVS datasets, only those objects that emit sound are given annotations. This dataset bias weakens the ability of visual models to identify all potential sounding objects [31, 37]. To better equip the visual networks for the AVS task, we incorporate a silent object-aware loss [31] into the training process. Additionally, we set all hyer parameters to the default settings of MaskFormer (Swin-base) and DETR (DETR-DC5 R101), including the backbone and the Transformer-based module. Both models are initially pre-trained on COCO. For further refinement, we fine-tune MaskFormer over five epochs and DETR over six epochs, both using the AdamW optimizer [34] at learning rates of 0.00006 and 0.0001, respectively.

Audio Branch. We employ VGGish accompanied by two subsequent multi-layer perceptrons (MLPs) as the audio branch. The training objective of the audio branch is Binary Cross Entropy Loss. The hyperparameters for VGGish are maintained at their standard settings, and we fine-tuned the module based on the version pre-trained on the VGGSound dataset. This whole branch is trained over 60 epochs using the AdamW optimizer, with a set learning rate of 0.001.

8.2. Modality Contribution Analysis

Table 3. Analyzing the distinct contributions of Strong Baseline's audio (A) and visual (V) branches: notably, the audio branch lacks spatial aspects, focusing evaluation metrics solely on temporal performance. For instance, m_tIoU measures label accuracy within the ground-truth temporal range. Conversely, the visual branch identifies vocalization periods by localizing the audible object's first and last frame appearances as its start and end times.

Modality		A	V	A&V
Ours (Mask_based)	m_tIoU	53.26	5.86	18.79
	m_vIoU	62.34	5.34	17.32
	m_tF	61.89	5.91	17.33
	m_vF	60.16	5.74	16.25
Ours (BBox_based)	m_tIoU	53.26	4.76	15.53
	m_vIoU	62.34	4.51	15.89
	m_tF	61.89	_	_
	m_vF	60.16	_	_

We further examine the impact of each modality based

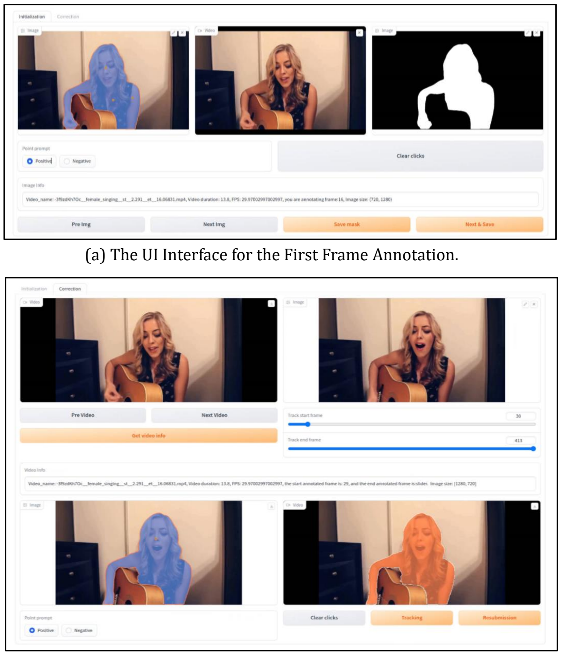

Image Info: Video_name: -3f9zdKh7Oc__female_singing__st__2.291__et__16.06831.mp4, Video duration: 13.8, FPS: 29.97002997002997, you are annotating frame:16, Image size: (720, 1280)

(a) The UI Interface for the First Frame Annotation.

Video Info: Video_name: -3f9zdKh7Oc__female_singing__st__2.291__et__16.06831.mp4, Video duration: 13.8, FPS: 29.97002997002997, the start annotated frame is: 29, and the end annotated frame is:slider. Image size: [1280, 720]

(b) The UI Interface for the Checking and Re-annotation.

Figure 9. The UI interface of our mask annotator tool.

on the strong baseline presented in Table 3. Notably, spatial localization is not involved when we solely employ the audio branch. Thus, evaluation metrics are exclusively focused on assessing model performance in the temporal dimension.

As suggested in Table 3, when only using audio modality as input, our model obtains competitive performance on all evaluation metrics relative to the AVE methods listed in Table 2. This implies that the prevalence of silent segments in the LU-AVS dataset limits the effectiveness of audio-visual interactions in current AVE methods. When we rely solely on the visual branch, we identify temporal localization by the frames where the target object first and last appears. As indicated in Table 2, the performance significantly declines across all metrics when solely using this branch. This is because the visual branch indiscriminately segments objects across all frames without considering sound emission, resulting in a significant reduction in both temporal and spatial accuracy.

Our findings emphasize the importance of audio guidance in the AVS task. Moreover, the results further demonstrate the intricacies of LU-AVS, including the high proportion of silent objects, multiple audible segments in a video, various sounding positions, and different duration lengths of audible segments, posing new challenges for AVS. This data complexity better reflects whether the method has truly achieved audio-visual localization, rather than fitting to prominent objects in the image, which would artificially inflate the model performance.