AuralSAM2: Enabling SAM2 Hear Through Pyramid Audio-Visual Feature Prompting (Supplementary Material)

Yuyuan Liu [1] Yuanhong Chen [2] Chong Wang [3] Junlin Han [1] Junde Wu [1]
Can Peng [1] Jingkun Chen [1] Yu Tian [4(✉)] Gustavo Carneiro [5]

[1] Department of Engineering Science, University of Oxford [2] Australian Institute for Machine Learning, Adelaide University
[3] Stanford University [4] University of Central Florida [5] University of Surrey

Outline

In this file, we supplement more content from the following aspects to support the findings and experimental results in the main body of the paper:

- **Section 1.1**: Training configurations, including backbone settings, transformer components, and loss weighting.
- **Section 1.2**: Data augmentation used throughout training.
- **Section 1.3**: Cross-modal fusion design, including projection layers and attention operations.
- **Section 1.4**: Contrastive learning setup, including sampling strategy and hyper-parameters.
- **Section 1.6**: Pseudocode for AuralFuser, covering multi-stage prompt construction and SAM2 decoding.
- **Section 1.5**: Re-implementation details of GAVS [14] and SAMA-AVS [5] using SAM2.
- **Section 3.7**: Additional prompt-engineering experiments comparing point/box prompts, adapter prompts, and our audio–visual prompting.
- **Section 3.1**: Results of applying AuralFuser to task-specific backbones (e.g., Swin-B).
- **Section 3.2**: Results of applying AudioCon to task-specific method CAVP [1].
- **Section 3.3**: Statistical evaluations across multiple seeds showing the stability of our results on AVSBench (V1m).
- **Section 3.4**: Ablation studies on incomplete-modality settings in the Ref-AVS dataset.
- **Section 3.5**: Analysis of failure cases, highlighting challenges in scenes with visually similar sounding objects.
- **Section 3.6**: Ablation study on input resolution, demonstrating robustness under reduced resolutions.
- **Section 3.7**: Evaluation of promptable capability based on human-in-the-loop interaction.
- **Section 3.8**: Analysis of computational cost, including memory usage and training time.
- **Section 3.9**: Evaluation on long video scenarios [4].
- **Section 4.1**: T-SNE visualisations showing improved clustering of audio–visual features.
- **Section 4.2**: Additional qualitative visualisations on AVSBench and Ref-AVS compared with other methods.

1. More Implementation Details

1.1. Hyper-parameter Configuration

Our method is based on SAM2 [11], utilizing the Hiera_base+ and Hiera_large backbones within the PyTorch framework, both of them remain frozen during training. We employ a batch size of one, where each batch consists of 5 frames for the V1s and V1m subsets in AVSBench [16], and 10 frames for the V2 subset in AVSBench [17] and the Ref-AVS [15] dataset. Training for datasets with 5-frame sequences is conducted with RTX 3090 GPU, whereas datasets with 10-frame sequences are trained with RTX A100 (40GB) GPU. We utilise learning rate equal to $1e^{-4}$ with a polynomial decay schedule, following $(1 - \frac{\text{iter}}{\text{max iter}})^{0.9}$ throughout the entire experiment. The number of training epochs is set to 180 for all experiments. Optimization is performed using the AdamW optimizer [8] with $\beta = (0.9, 0.999)$ and a weight decay of 0.01, without applying any gradient clipping. Our method processes the visual modality and the audio-language modalities using the self-attention mechanism within the transformer blocks before cross-modal fusion. For the visual modality, we employ 9 transformer blocks with the same structure as in PVT [13]. For the audio-text modality branch, we utilize 3 transformer blocks and follow standard practices [2, 11] for self-attention. In both modalities, the self-attention configuration consists of 4 attention heads with a dropout rate of 0.1. In terms of the SAM2 loss, which includes l_{focal}, l_{dice}, l_{iou} and l_{occ} in l_{SAM2}, we apply weight ratios of 20:1:1:1. Following the original SAM2 paper's configuration, we penalise only the best-predicted segmentation mask, which is determined as the one with the minimal loss based on $l_{focal} + l_{dice}$. During inference, we use the best-predicted IoU to select the class-agnostic mask from the set of predicted masks. Please note that we do

not apply any post-processing techniques such as test-time augmentation (TTA), largest connected components, or internal hole filling in our experiments. To encapsulate the semantic information in the AVSBench (V2) [17] dataset, we employ the Stepping-Stone [9] method to train class tokens using pre-predicted class-agnostic masks generated by our approach. We fine-tune the officially released code for an additional 40 epochs and report the final results using the same evaluation metrics. All resources associated with this paper, including the code, checkpoints, and training logs, will be made publicly available.

1.2. Augmentation Configuration

We apply color jittering both at the video level and frame level, along with random horizontal flipping and random darkred!50scale transformation with a probability of 0.1, following the SAM2 [11] training pipeline. We do not use random cropping; instead, all input frames are resized to a resolution of $1024 \times 1024 \times 3$ throughout the experiments. Additionally, no augmentations are applied to the audio data.

1.3. Cross-modalities Fusion Details

Our cross-modal fusion is adapted from TPAVI [16] and incorporates an additional cross-attention for audio-language modalities, enabling the two-way cross-fusion. Specifically, we have input visual modalities represented as $\mathbf{r}_v \in \mathbb{R}^{B\times H'\times W'\times L}$ and input audio-text modalities with shape $\mathbf{r}_c \in \mathbb{R}^{(B+N^t)\times L}$, where B is the batch size, $H' \times W'$ is the resolution of the feature map, N^t is the number of words in the sentence, and L is the latent dimension. We use `conv3D` as the projection layer for the visual modality and `conv1D` for the audio-text modalities. Then we can have $\{q_\text{v}, k_\text{v}, v_\text{v}\} \in \mathbb{R}^{(B\times H'\times W')\times L'}$ for the visual modality, and $\{q_\text{c}, k_\text{c}, v_\text{c}\} \in \mathbb{R}^{(B+N^t)\times L'}$ for other modalities, where L' is compressed dimension. After that, we calculate the cross-modality fusion as following:

$$\mathbf{r}_v = \text{softmax}\left(\frac{q_\text{v}k_\text{c}^\top}{\sqrt{d}}\right)v_\text{c} \quad \mathbf{r}_c = \text{softmax}\left(\frac{q_\text{c}k_\text{v}^\top}{\sqrt{d}}\right)v_\text{v}, \tag{1}$$

where d is the normalise value to avoid large magnitudes. Followed by batch normalisation and a MLP that reduces the dimension from L' back to L, $\mathbf{r}_v$ and $\mathbf{r}_c$ continue the training pipeline as described in Eq. (5) of the main paper.

1.4. Contrastive learning Details

We utilise a 3-layer MLP to project the latent embeddings from the audio and visual modalities into a 64-dimensional space, respectively. For the pyramid multi-scale visual features, we randomly select 512 visual embedding samples from each scale in every frame. Except for AVSBench (V1s)[16], we apply the InfoNCE[10] loss only on the first frame during training, as it is the only frame with available labels. Following [6, 7, 12], we perform hard and easy sample mining based on the ground truth. Embeddings corresponding to correctly predicted results are treated as easy samples, while embeddings associated with incorrectly predicted results are considered hard samples, maintaining a balanced 1:1 ratio. We adopt the default temperature value of 0.1 from [12] without further fine-tuning and we don't apply any weight to the contrastive loss in Eq. (8) main paper.

Algorithm 1 Pseudocode of AuralFuser (PyTorch-like)

```
# Inputs: frames, spectrogram;
# Backbones: SAM2(fixed), VGGish
# Modules: AuralFuser
# Output: segmentation masks

# prepare multi-modal features.
a_feats = VGGish.encoder(spectrogram)
# v_feats ∈ {v_feat1, v_feat2, v_feat3}
v_feats = SAM2.encoder(frames)

# prepare position encodings.
a_pos = PosEnc1D(length=a_feat)
v_pos = PosEnc2D(length=v_feat)

# prepare dense & sparse prompts.
for each stage k = 1 to 3 do
    # p is the patch size used to align the different
    # dimensions of v_feat_k.
    p = (4-k) ** 2    # lateral layer when k=3.
    d_prpt_k = PatchEmbed_k(v_feat_k, p) + v_pos
    s_prpt_k = a_feat_k + a_pos
    if k > 1 then
        # feature pyramid collection.
        d_prpt_k = Pyramid_k(d_prpt_k, d_prpt_k-1)
    end if
    d_prpt_k, s_prpt_k = Fusion_k(d_prpt_k, s_prpt_k)
end for
# (sparse + dense) prompts
prompts = (s_prpt_1,2,3, d_prpt_1,2,3)
# calculate final outputs with original visual_feats.
seg_masks, ious, occ = \
            SAM2.Decoder(visual_feats, prompts)
# decide using best IoU and occlusion check
output = seg_masks[argmax(ious)][occ]
return output
```

1.5. Re-implementation of Other Works.

We directly apply the SAM2 model to the GAVS [14] and SAMA-AVS [5] approaches, replacing their original SAM

Table 1. **Prompt Engineering with Audio** in the AVSBench (V1m) [16] dataset with Hiedra_base+ backbone. We use points and boxes generated from ground truth to simulate real-world prompting practices. The green rows represent SAM2-based methods that receive visual prompts while retaining their adapters for promptable segmentation. The blue rows indicate the AVS results of SAM2-based methods, while the yellow rows show results obtained by ensemble learning, combining both AVS and SAM2's original promptable segmentation results. The FPS represents the number of frames processed per second, and the best segmentation results highlighted in red.

Methods	Prompts	$\mathcal{M}_{\mathcal{J}}$	$\mathcal{M}_{\mathcal{F}}$	FPS
SAM2 [11]	points	64.67	72.15	17.8
	box	68.85	76.52	17.4
	mask	75.73	81.54	16.9
	points box	72.64	79.56	17.2
GAVS [14]	points box	69.34	77.32	16.9
SAMA-AVS [5]	points box	70.25	78.54	17.0
Ours	points box	72.64	79.56	17.2
GAVS [14]	audio	68.13	79.07	14.8
SAMA-AVS [5]	audio	67.70	78.93	16.3
Ours	audio	72.04	81.46	16.4
GAVS [14] (w/ SAM2)	audio points box	71.70	81.94	8.7
SAMA-AVS [5] (w/ SAM2)	audio points box	69.74	80.97	9.9
Ours (w/ SAM2)	audio points box	74.26	83.58	14.1

model. In AVSBench [16, 17], for GAVS [14], we inject Multi-Layer Perceptron (MLP) adapters after the 9th layer of the image encoder, with an intermediate latent dimension of 128. Additionally, we insert adapters into the mask decoder with the same latent dimension during the two-way cross-attention process. For SAMA-AVS [5], we expand the intermediate dimension to 512 for each adapter, following its setup. In both methods, the adapter outputs are directly added to the image features during the forward pass of the image encoder. In Ref-AVS [15], we further employ cross-attention to fuse the adapter outputs with encoded textual features, facilitating cross-modal fusion between the audio-language modalities.

1.6. Pseudo-Code of AuralFuser

We present the pseudocode of our proposed AuralFuser in Algorithm 1, where the input consists of video frames and spectrograms. After extracting features from both modalities using fixed backbones, feature-level prompts are generated at each stage through cross-modal fusion (as shown in Supp. Section 1.3), leveraging the pyramid of visual features. These prompts are divided into sparse prompts, primarily derived from audio and capturing the global context of the current frame, and dense prompts, mainly derived from visual features and encoding fine-grained pixel-level details. Both sets of prompts are then fed into the decoder.

2. Prompting Engineering

We provide additional details on prompt engineering based on the Hiera_base+ backbone in AVSBench (V1m)[16], as shown in Tab.1. In the first four rows, we report the visual prompt results for SAM2, including four uniformly generated points and boxes derived from the ground truth mask. Since pixel-level labeled masks are challenging to obtain in practice, we use only points and boxes in this experiment. The following three green rows present the promptable segmentation results of SAM-based AVS methods [5, 14] using point and box visual prompts. We observe a decline in segmentation performance for the adapter-based methods, with GAVS [14] showing a 3.3% drop in $\mathcal{M}_{\mathcal{J}}$ and SAMA-AVS [5] experiencing a 1.79% decrease. This decline occurs because the injected adapters modify image features, reducing SAM2's original generalisation capability. Next, we compare AVS results in the blue rows , where our method achieves the best performance and efficiency. For instance, compared to GAVS [14], our approach improves $\mathcal{M}_{\mathcal{J}}$ results by 3.31% while also achieving an 1.6 FPS increase. This improvement is due to the fact that the numerous adapters within the image encoder can slow down inference speed. Finally, in the last three yellow rows , our method successfully enhances SAM2's promptable segmentation performance, achieving a Jaccard score of 1.62 with an efficiency cost of 3.5 FPS. This remains significantly faster than GAVS [14] at 8.7 FPS and SAMA-AVS [5] at 9.9 FPS.

3. Experiments (additional)

3.1. Applying AuralFuser to Task-specific Models

In Tab. 2, we evaluate our method using a task-specific backbone (Swin-B) on the AVSBench dataset [17]. We construct the multimodal pyramid by extracting feature embeddings from Stages 1, 2, and 3 of Swin-B after each patch-merging layer and fuse these with audio embeddings to support cross-modal learning. All other training settings, such as learning rate and batch size, mirror those in the main paper. Given the smaller model scale, we retrained the full architecture end-to-end, similar to prior task-specific methods. The results demonstrate improvements of +1.7% and +1.4% in mean Jaccard and F-score on AVSBench (V1s), as well as +1.4% and +1.4% on AVSBench (V1m), respectively. These gains are directly attributable to our AuralFuser design: leveraging a multi-scale feature pyramid ensures early-stage fine-grained detail and later-stage contextual information are both audio-enhanced. Additionally, our AudioCon alignment encourages visual embeddings to adhere closely to audio prototypes across these scales, reinforcing audio's guiding role throughout the model hierarchy.

Table 2. **Ablation studies** with task-specific model (Swin-b) on AVSBench datasets. The best results are in red.

Method	Backbone	AVSBench (V1s)		AVSBench (V1m)	
		$\mathcal{M}_{\mathcal{J}}$	$\mathcal{M}_{\mathcal{F}}$	$\mathcal{M}_{\mathcal{J}}$	$\mathcal{M}_{\mathcal{F}}$
Step.Stone	VGG Swin-b	83.2	91.3	67.3	77.6
Ours	VGG Swin-b	84.9	92.0	68.7	79.0

Table 3. **Ablation studies** that applied our contrastive learning on the CAVP [1] method with the results reported based on their official Github. The best results are in red.

Method	Backbone	AVSBench (V1s)		AVSBench (V1m)	
		$\mathcal{M}_{\mathcal{J}}$	$\mathcal{M}_{\mathcal{F}}$	$\mathcal{M}_{\mathcal{J}}$	$\mathcal{M}_{\mathcal{F}}$
CAVP	Resnet50	78.8	88.9	55.8	67.1
CAVP (w/AudioCon)	Resnet50	79.6	89.3	57.3	68.2

Table 4. **Statistical results** of our method using the Hiera_base+ backbone on the AVSBench (V1m) [16] dataset. Red values indicate those reported in the main paper. The last two columns present the mean (μ) and stand deviation (σ) to assess the robustness and reproducibility of our method.

Seed (#)	555	666	777	888	999	μ	σ
$\mathcal{M}_{\mathcal{J}}$	71.94	72.04	72.13	71.88	71.92	71.98	0.09
$\mathcal{M}_{\mathcal{F}}$	81.39	81.46	81.52	81.32	81.36	81.41	0.07

3.2. Applying AudioCon to Task-specific Models

In Table 3, we apply **AudioCon** to the CAVP method (with a ResNet-50 backbone) on AVSBench. Following our main implementation, AudioCon enforces positive alignment exclusively between visual embeddings and their corresponding audio prototypes, significantly reducing pixel-level visual–visual reinforcement that leads to modality dominance. As a result, we observe consistent gains across both AVSBench variants: a Jaccard improvement of +0.8% and F-score improvement of 0.4% on AVSBench (V1s), and +1.5% and +1.1%, respectively, on AVSBench (V1m). These improvements demonstrate AudioCon's effectiveness in amplifying audio influence within contrastive learning: by promoting audio-guided alignment, AudioCon ensures auditory signals contribute meaningfully to segmentation performance, thus mitigating the limitations of purely visual supervision.

3.3. Statistical tests for AVSBench (V1m)

We evaluate the robustness of our method on the AVSBench (V1m)[16] subset by running five independent implementations using the Hiera_b+ backbone, as reported in Tab.4. All runs are executed on 4×3090 GPUs with identical software environments (e.g., CUDA and cuDNN versions), while varying only the random seed (listed in the first row). Across these five runs, our method achieves a mean of 71.98±0.09 for $\mathcal{M}_{\mathcal{J}}$ and 81.41 ± 0.07 for $\mathcal{M}_{\mathcal{F}}$, demonstrating the stability and reproducibility of our final results.

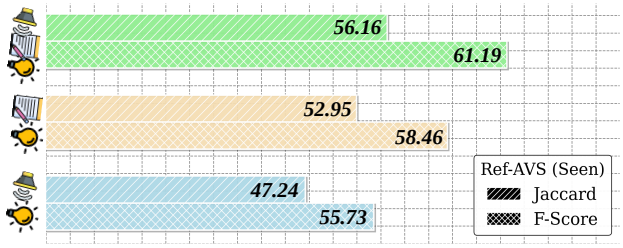

Figure 1. **Ablation Studies on missing modalities** in Ref-AVS (Seen subset) [15] using Hiera_l backbone, evaluating the importance of audio, language and visual modalities.

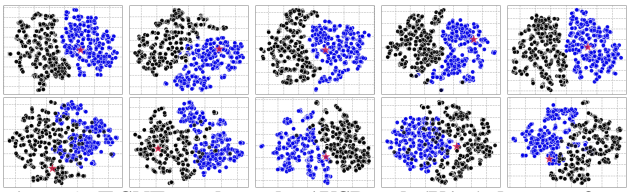

Figure 2. **T-SNE result on the AVSBench (V1m) dataset of our method on Hiera_L** (refer to Tab.5, main paper). The first row shows results from AudioCon, while the second row corresponds to SupCon. Each column represents a frame from the same video sequence. The red star denotes the projected audio embedding, while the blue and black circles represent the projected foreground and background pixel-level visual features.

Figure 3. Top: GT overlay. Bottom: Hiera_l results with IoU (red).

Table 5. **Ablation of Image Resolution** in AVSBench-V1m [16].

Method	1024x1024		512x512		224x224	
	$\mathcal{M}_{\mathcal{J}}$	$\mathcal{M}_{\mathcal{F}}$	$\mathcal{M}_{\mathcal{J}}$	$\mathcal{M}_{\mathcal{F}}$	$\mathcal{M}_{\mathcal{J}}$	$\mathcal{M}_{\mathcal{F}}$
Ours (Hiera_L)	75.58	84.12	74.04	83.79	71.49	80.58

3.4. Ablation Studies on Missing Modalities

In Fig. 1, we conduct ablation studies on the Ref-AVS (Seen subset) [15] to assess the contribution of audio, text, and visual modalities. Using only audio-visual modalities, our method achieves 47.24% in Jaccard and 55.73% in F-score. Incorporating language-visual modalities improves performance by 5.71% in Jaccard. When all three modalities are combined, the results further increase by 3.21% and 2.34% in Jaccard and F-score, respectively.

3.5. Failure Cases Analysis

Failure cases mainly occur in scenes with sounding objects in *similar* appearances (Fig.3). This is because VGGish collapses stereo audio into mono log-mel features and does not preserve spatial localisation cues, making such cases hard to disambiguate.

Table 6. **Real human-in-the-loop evaluation.** We report mean Jaccard over 30 video clips with 3 users from AVSBench-V1m.

Prompt Resp.	**R1**	**R2**	**R3**	**R@$\mathcal{M}_\mathcal{J} \geq 0.9$**
SAM2	69.90 ± 3.36	88.53 ± 1.80	92.23 ± 1.25	2.67 ± 0.58
AuralSAM2	**75.23** ± 1.96	**90.47** ± 0.63	**93.37** ± 0.72	**2.00** ± 0.05
Conflict Handling	**Conflict Prompt@R1**			**Recovery Prompt@R2**
SAM2	6.33 ± 2.51			69.34 ± 2.87
AuralSAM2	**52.96** ± 1.36			**73.96** ± 2.13

Table 7. **Results on LU-AVS.** m_tIoU and m_tF assess temporal overlap accuracy, while m_vIoU and m_vF evaluate joint temporal and pixel-level segmentation performance.

Method	m_tIoU ↑	m_tF ↑	m_vIoU ↑	m_vF ↑
ST-AG [CVPR'24]	18.76	17.33	17.32	16.25
Ours [Hiera_l]	**42.63**+23.87	**43.94**+26.61	**41.09**+23.77	**42.17**+25.92

Table 8. **Budget Analysis on AVSBench-V1m** via 4090 GPU.

Training Cost	GAVS [b+]	SAMA-AVS [b+]	Ours [b+]	Ours [l]
Peak Memory (GB)	19.14	20.09	**18.94**	20.61
Training Time (h)	2.9	3.2	**2.6**	3.7

3.6. Ablation Study on Input Resolution

All SAM/SAM2-based AVS methods included in our comparison operate at a resolution of 1024×1024. To ensure a fair comparison, we follow the same setting in our main experiments. In addition, we evaluate our method under lower input resolutions, as reported in Tab. 5. The results show that our approach maintains strong performance even when the resolution is reduced, demonstrating its robustness to input scaling.

3.7. Promptable Capability

We evaluate the promptable capability of our method under both standard prompting settings and human-in-the-loop interaction. Specifically, Tab. 6 (main paper) and Supp. Sec. 2 report results on audio-guided segmentation combined with point and box prompts, measuring both task performance (accuracy) and interaction latency (FPS). To further assess practical prompt responsiveness, we conduct real human-in-the-loop experiments, as summarized in Tab. 6. We measure performance across multiple interaction rounds (R1–R3), as well as the ability to handle incorrect prompts followed by corrective inputs. Three independent participants evaluate the same set of 10 video clips. Following the SAM2 protocol, prompts are provided only on the first frame of the sounding object. The results demonstrate that our method preserves the promptable capability of SAM2 while incorporating audio guidance, maintaining both effective interaction and stable performance under iterative user inputs.

3.8. Computational Cost and Scalability

We report peak memory usage and training time under identical settings with SAM2 in Tab. 8. When scaling to Hiera_l, the additional computational cost mainly arises from the larger backbone forward pass, while the AuralFuser module remains unchanged and introduces negligible overhead.

3.9. Long Video Scalability.

We further evaluate our method on the LU-AVS [4] benchmark without any architectural modification. As shown in Tab. 7, our method achieves more than $2\times$ improvement across all metrics, demonstrating strong scalability to long video scenarios.

4. Visualisations

4.1. T-SNE Visualisation

The previous methods, such as SupCon [3] used in CAVP [1] often causes the model to overfit to dominant visual features due to their much larger quantity, leading to false detections of visually frequent but silent objects. This issue is evident in Fig. 2 (second row), where pixels of the sounding object (blue circles) are mixed with background pixels (black), and the corresponding audio embedding is not clearly clustered. In contrast, AudioCon forms positive pairs solely between visual and audio features, without constraints from other visual features, and applies this across all levels of hierarchical prompts. This encourages better modeling of the projected audio-visual feature distribution, as shown in Fig. 2 (first row).

4.2. Qualitative Visualisation

We present qualitative visualisation results comparing our method with other adapter-based approaches, GAVS [14] and SAMA-AVS [5]. Specifically, Figures 4 and 5 illustrate the outputs in multimodal scenarios involving audio, language, and visual modalities within the Ref-AVS (seen)[15] subset, while Figures 6 and 7 show results for its unseen subset. In the AVSBench[16, 17] dataset, which incorporates audio-visual modalities, we visualize results for V1s (single-sounding source data) in Figures 8 and 9, V1m (multiple-sounding sources) in Figures 10 and 11, and V2 (more complex scenarios) in Figures 12 and 13. Overall, with the same Hiera_base+ backbone, our method achieves superior visualizations, with further improvements when adopting the Hiera_large backbone.

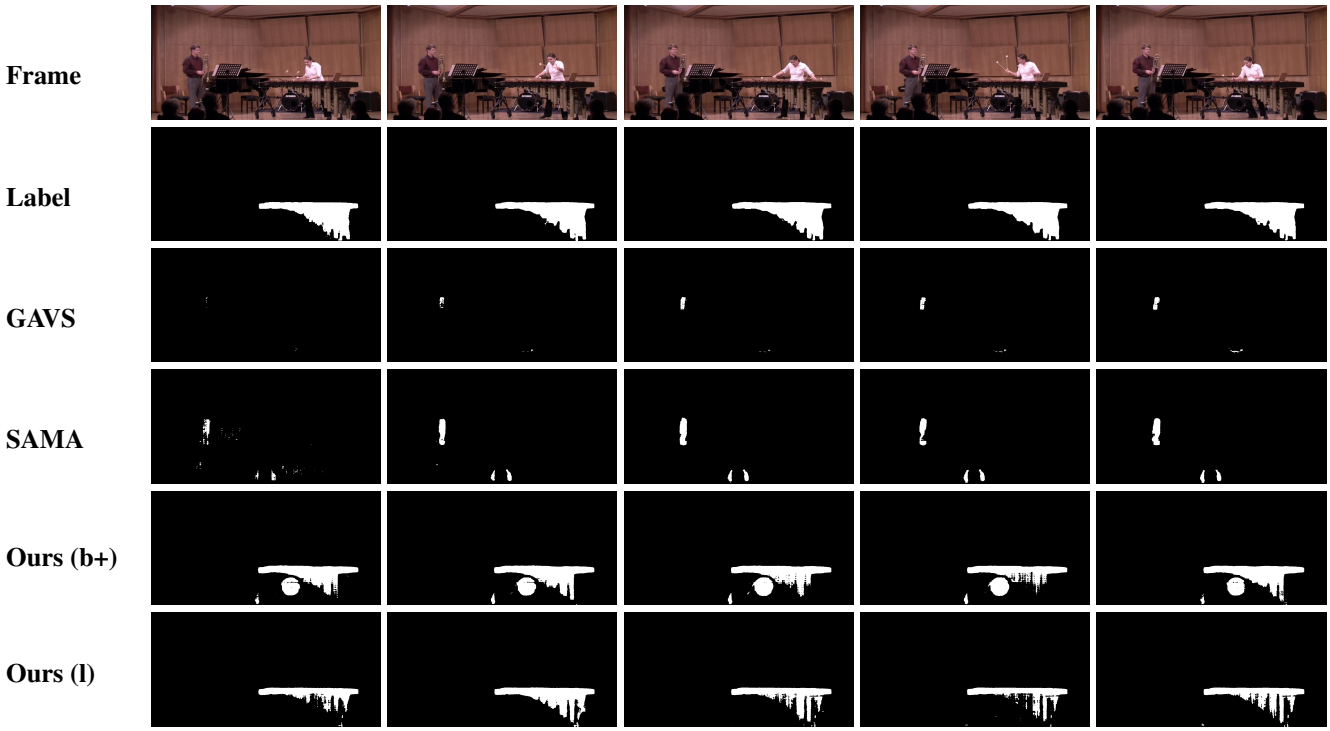

Figure 4. 'The object making a sound by being played by the woman.' from Ref-AVS (seen) [15]

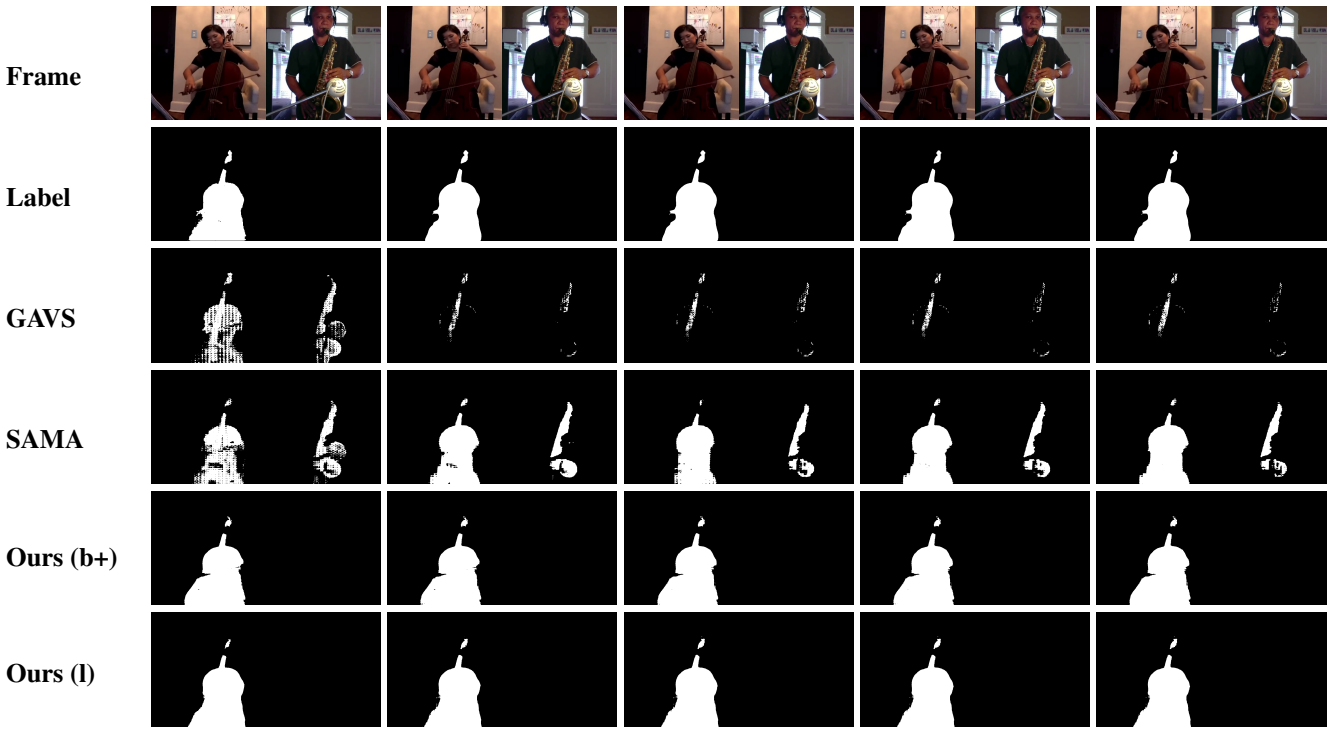

Figure 5. 'The object producing sound under the manipulation of the individual on the left.' from Ref-AVS (seen) [15]

Figure 6. 'The object making the longest sound duration.' from Ref-AVS (unseen) [15]

Figure 7. 'The object that keeps making sound at all times.' (from Ref-AVS (unseen) [15])

Figure 8. case (a) from AVSBench (V1s) [16]

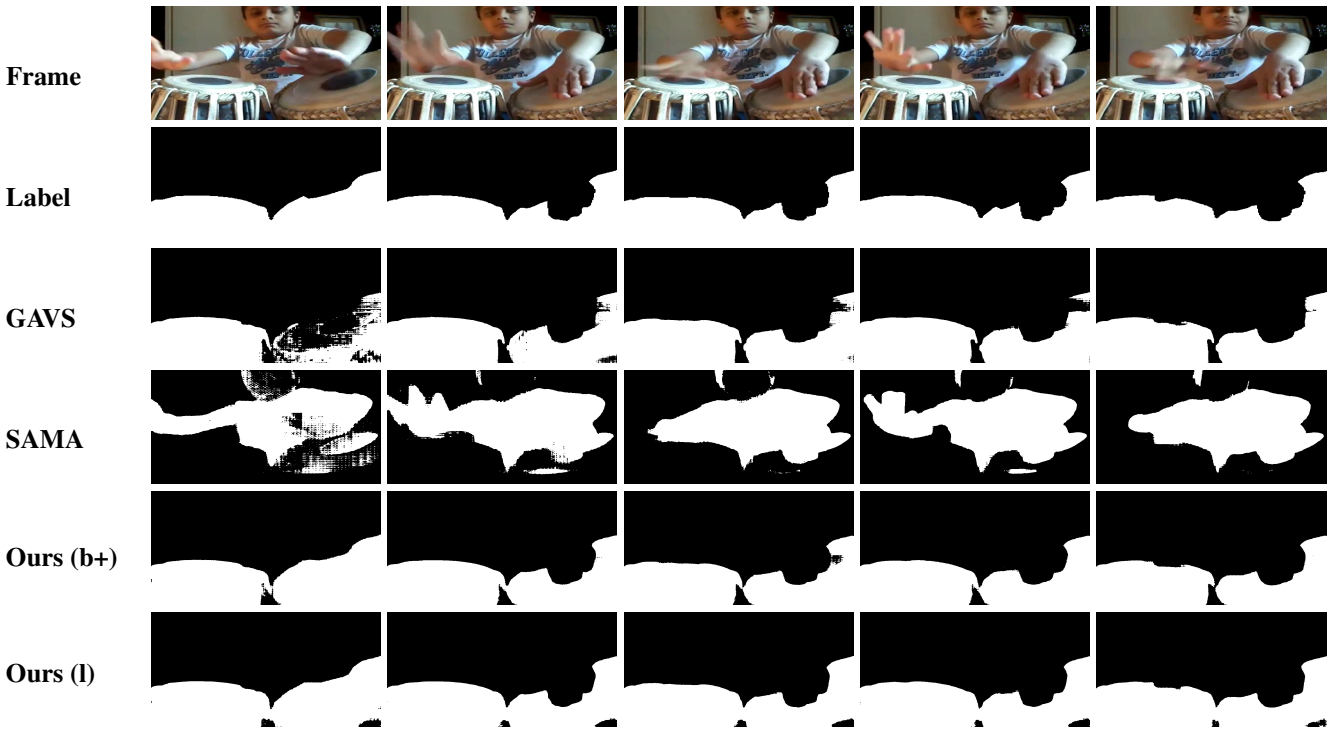

Figure 9. case (b) from AVSBench (V1s) [16]

Figure 10. case (a) from AVSBench (V1m) [16]

Figure 11. case (b) from AVSBench (V1m) [16]

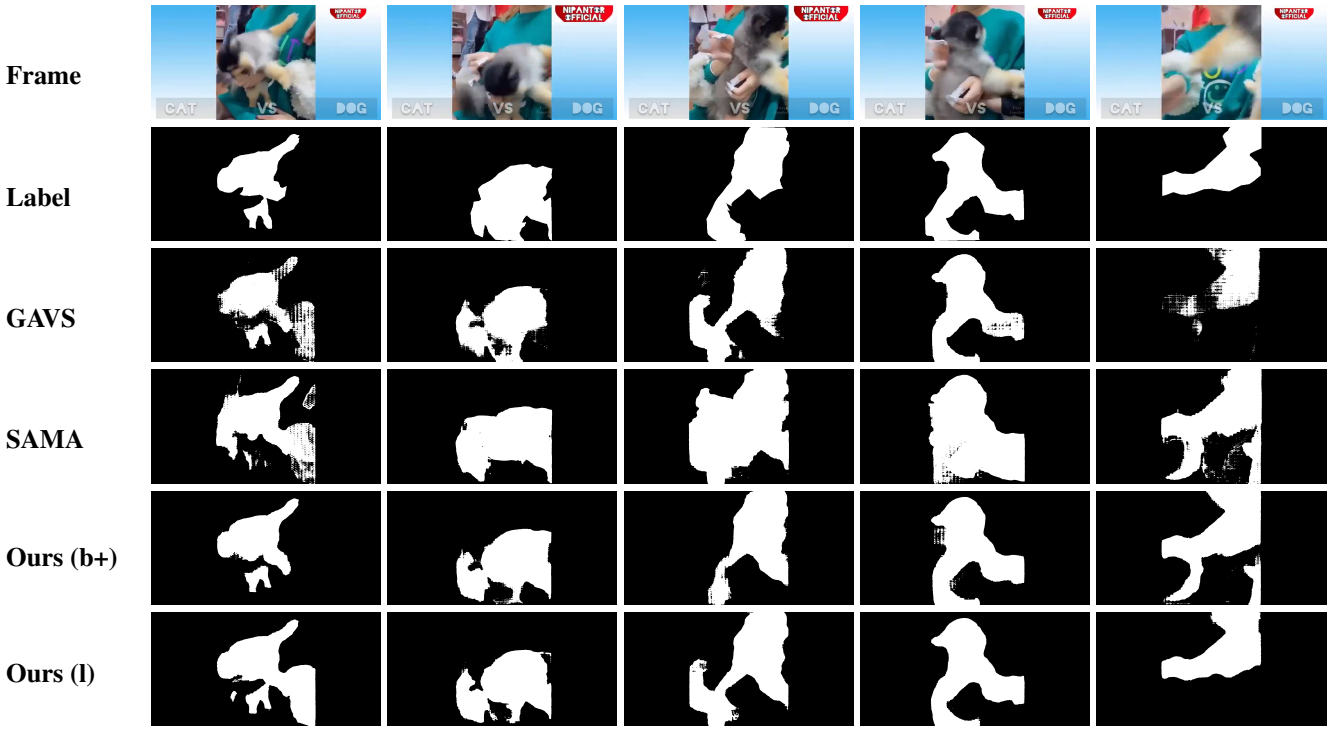

Figure 12. case (a) from AVSBench (V2) [17]

Figure 13. case (b) from AVSBench (V2) [17]

References

[1] Yuanhong Chen, Yuyuan Liu, Hu Wang, Fengbei Liu, Chong Wang, Helen Frazer, and Gustavo Carneiro. Unraveling instance associations: A closer look for audio-visual segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26497–26507, 2024. 1, 5

[2] Yuanhong Chen, Chong Wang, Yuyuan Liu, Hu Wang, and Gustavo Carneiro. Cpm: Class-conditional prompting machine for audio-visual segmentation. In *European Conference on Computer Vision*, pages 438–456. Springer, 2025. 1

[3] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020. 5

[4] Chen Liu, Peike Patrick Li, Qingtao Yu, Hongwei Sheng, Dadong Wang, Lincheng Li, and Xin Yu. Benchmarking audio visual segmentation for long-untrimmed videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22712–22722, 2024. 1, 5

[5] Jinxiang Liu, Yu Wang, Chen Ju, Chaofan Ma, Ya Zhang, and Weidi Xie. Annotation-free audio-visual segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5604–5614, 2024. 1, 2, 3, 5

[6] Yuyuan Liu, Choubo Ding, Yu Tian, Guansong Pang, Vasileios Belagiannis, Ian Reid, and Gustavo Carneiro. Residual pattern learning for pixel-wise out-of-distribution detection in semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1151–1161, 2023. 2

[7] Yuyuan Liu, Yuanhong Chen, Hu Wang, Vasileios Belagiannis, Ian Reid, and Gustavo Carneiro. Ittakestwo: Leveraging peer representations for semi-supervised lidar semantic segmentation. In *European Conference on Computer Vision*, pages 81–99. Springer, 2024. 2

[8] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[9] Juncheng Ma, Peiwen Sun, Yaoting Wang, and Di Hu. Stepping stones: A progressive training strategy for audio-visual semantic segmentation. *IEEE European Conference on Computer Vision (ECCV)*, 2024. 2

[10] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[11] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 1, 2, 3

[12] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7303–7313, 2021. 2

[13] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. 1

[14] Yaoting Wang, Weisong Liu, Guangyao Li, Jian Ding, Di Hu, and Xi Li. Prompting segmentation with sound is generalizable audio-visual source localizer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5669–5677, 2024. 1, 2, 3, 5

[15] Yaoting Wang, Peiwen Sun, Dongzhan Zhou, Guangyao Li, Honggang Zhang, and Di Hu. Ref-avs: Refer and segment objects in audio-visual scenes. In *European Conference on Computer Vision*, pages 196–213. Springer, 2025. 1, 3, 4, 5, 6, 7

[16] Jinxing Zhou, Jianyuan Wang, Jiayi Zhang, Weixuan Sun, Jing Zhang, Stan Birchfield, Dan Guo, Lingpeng Kong, Meng Wang, and Yiran Zhong. Audio–visual segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVII*, pages 386–403. Springer, 2022. 1, 2, 3, 4, 5, 8, 9

[17] Jinxing Zhou, Xuyang Shen, Jianyuan Wang, Jiayi Zhang, Weixuan Sun, Jing Zhang, Stan Birchfield, Dan Guo, Lingpeng Kong, Meng Wang, et al. Audio-visual segmentation with semantics. *arXiv preprint arXiv:2301.13190*, 2023. 1, 2, 3, 5, 10