Seeing Speech and Sound: Distinguishing and Locating Audio Sources in Visual Scenes

Supplementary Material

The contents in this supplementary material are as follows:

Details on Extended IS3 Dataset (Section 7), Re-visiting Heads (Section 8), Additional Ablation Results (Section 9), Computational Overhead (Section 10), Additional Qualitative Results (Section 11), Performance Discrepancy with DenseAV (Section 12) and Implementation Details (Section 13).

7. Details on Extended IS3 Dataset

We extend the IS3 dataset [36] to enable simultaneous grounding of mixture of audios. The dataset is originally designed for interactive sound source localization, consisting of 3,420 images, each paired with two general sound samples corresponding to two visible objects in the scene. We generate speech samples for each visible object in each image. These speech samples, created using the Google gTTS API, read the class categories of the two visible objects. Then each image contains two objects, each associated with one sound and one speech sample. We form triplets by combining the image with the sound from one object and the speech from the other, and vice versa. The sound and speech samples of a triplet are then mixed together to form a single combined auditory input. This extension enables simultaneous grounding of mixed audio types which requires disentangling overlapping auditory inputs—sound and speech—and aligning them with the correct visual objects. The Extended IS3 dataset thus serves as a comprehensive benchmark for evaluating the capability on audio-visual interactions in real-world scenarios.

8. Re-visiting Heads

As outlined in our architecture, the audio encoder includes two specialized heads: one for speech and one for sound. Additionally, in Section 4.3.2 of the main paper, we discuss different evaluation approaches for these heads, namely *Specialized Heads* and *Total Head*. Here, we would like to re-emphasize that the primary focus of the evaluation should be on the specialized heads, depending on the target task. The *Total Head*, introduced by Hamilton *et al*. [9], serves as an alternative evaluation approach. In summary, tasks such as segmentation and retrieval on the Places dataset should use the *Speech Head*, while all tasks on the AudioSet dataset should utilize the *Sound Head*.

As shown in Table 3 and Table 5 of the main paper, the *Total Head* approach exhibits performance degradation on benchmarks such as AudioSet and Places. This performance gap is especially evident in Table 5, where the mixed audio scenario highlights the limitations of aggregating unrelated similarity volumes (*Total Head* approach). The inclusion of irrelevant information introduces noise, leading to considerably worse results compared to the *Specialized Head* approach, which focuses solely on the relevant audio type and achieves better performance. Similarly, as mentioned in the main paper (Section 4.3.4), CAV-MAE performs better than ours in the *Total Head* for I→A on AudioSet. However, the target for this task should be the *Sound Head*, and the performance degradation from *Sound Head* to *Total Head* can serve as evidence of disentanglement ability of our model, as the *Total Head* introduces noise from unrelated sub-modalities.

To highlight the scenario where the *Total Head* is particularly useful, we conducted an additional novel task on the Extended IS3 dataset. This task involves performing retrieval using audio inputs that combine speech and general sounds, along with images containing two related objects. In this case, using the *Total Head* is more appropriate as it better reflects the characteristic of the dataset and the task. We present results in Table 8 by comparing head selection methods. *Total* uses summation across sub-modalities:

$$S_{sum}(\mathbf{a}, \mathbf{v}) = \sum_{k} \left(S(\mathbf{a}, \mathbf{v}) \right), \tag{12}$$

where k refers the number of audio types. *Sound Head* and *Speech Head* directly apply the head selection approach in (4). *Total Head* evaluation outperforms both specialized heads on the Extended IS3 dataset by approximately 10% or more.

	Retrieval R@10					
	Total		Sound		Speech	
Method	**I→A**	**A→I**	**I→A**	**A→I**	**I→A**	**A→I**
DenseAV [9]$_{\text{CVPR24}}$	25.2	11.7	14.7	9.7	19.0	9.9
Ours	**29.7**	**25.0**	**18.3**	**11.9**	**19.8**	**16.8**

Table 8. **Cross-modal retrieval task on Extended IS3.**

9. Additional Ablation Results

In Section 4.4 of the main paper, we presented ablation results to evaluate the impact of our audio-visual alignment objectives—correspondence and disentanglement. Here, we provide additional results on the retrieval task for both clean and mixed audio cases to present a more comprehensive analysis. Results are in Table 9 and Table 10.

Firstly, as shown in Table 9, omitting the disentanglement loss causes the *Speech Head* to perform numerically higher (indicating worse performance) on AudioSet by approximately 30%, and the *Sound Head* to perform worse on Places by over 20% in clean retrieval. This suggests that the opposite heads are being activated by the incorrect audio type, which is undesirable. A similar trend is observed in the mixed retrieval results in Table 10, although the performance gap is slightly smaller. These findings indicate that without the disentanglement loss, the heads fail to specialize effectively for their intended roles. Secondly, we examine the impact of omitting the correspondence loss by evaluating the specialized head for each dataset. In both Table 9 and Table 10, performance drops by over 15% on AudioSet and ~3% on Places, confirming that L_{cor} effectively enables audio-visual matching necessary for cross-modal retrieval. It is noteworthy that the model trained only with L_{cor} struggles with mixed audio, as L_{dis} provides robustness against noise from the opposite audio type due to efficient disentanglement.

	Clean Retrieval R@10											
	AudioSet						Places					
	Total ↑		Sound ↑		Speech ↓		Total ↑		Sound ↓		Speech ↑	
Method	I→A	A→I	I→A	A→I	I→A	A→I	I→A	A→I	I→A	A→I	I→A	A→I
L_{cor}	**48.6**	**47.4**	34.0	35.8	38.3	38.7	**92.3**	91.9	23.7	42.1	92.0	92.5
L_{dis}	29.2	28.5	30.9	32.0	4.7	**7.1**	91.5	90.5	**2.1**	**6.3**	91.1	91.2
Ours	45.5	46.6	**51.2**	**50.0**	**3.7**	8	92.0	**93.1**	2.3	10.5	**94.0**	**94.0**

Table 9. **Cross-modal retrieval task on Places and AudioSet.**

	Mixed Retrieval R@10											
	AudioSet						Places					
	Total ↑		Sound ↑		Speech ↓		Total ↑		Sound ↓		Speech ↑	
Method	I→A	A→I	I→A	A→I	I→A	A→I	I→A	A→I	I→A	A→I	I→A	A→I
L_{cor}	14.0	11.4	11.3	9.6	7.5	9.4	49.8	51.2	7.8	21.4	52.1	53.3
L_{dis}	**19.4**	19.9	29.3	28.3	3.6	3.1	76.5	80.8	1.7	3.2	82.7	83.6
Ours	17.8	**20.3**	**44.3**	**42.6**	**1.6**	**1.3**	**79.0**	**82.8**	**1.4**	**1.4**	**87.3**	**86.2**

Table 10. **Cross-modal retrieval task on Places and AudioSet with mixture of audios.**

10. Computational Overhead

Model	DenseAV	Ours
FLOPs(G)	4151.25	12002.54

Table 11. **Computational overhead during training.**

Our approach uses multiple forward passes through the image and audio encoders to process clean audios, their paired images, and mixed audio inputs (Figure 2), increasing overhead during training compared to DenseAV [9] (Table 11). However, this overhead is only present during training, as the test phase is a single forward pass.

11. Additional Qualitative Results

11.1. Segmentation Benchmarks

Due to space constraints in the main paper, we included only a limited number of qualitative results from the Simultaneous Segmentation experiment. In this supplementary material, we provide additional qualitative results for the standard segmentation task, including Sound-Prompted Semantic Segmentation, Speech-Prompted Semantic Segmentation, and Simultaneous Segmentation on the Extended IS3 dataset, as shown in Figure 4, Figure 5, and Figure 6, respectively.

11.2. Real-world Scenarios

The examples in Figure 7 qualitatively compare our model with DenseAV in real-world scenarios from YouTube. When sound and speech overlap, our model grounds both the sound source and the object mentioned in speech more robustly. Our model demonstrates strong performance not only on evaluation dataset composed of TTS-generated speech but also on real-world speech samples.

11.3. Failure Cases

We present two failure cases: (1) When the interacted object is too small or occluded, the model may capture both the object and the person (Figure 8, left). (2) When additional background noise or music increases the complexity of the audio, it makes localization harder (Figure 8, right). Despite these challenges, our model remains more robust than DenseAV.

12. Performance Discrepancy with DenseAV

Our approach builds on DenseAV [9] by introducing joint learning objectives (Mix-and-Separate approach), while strictly following the official DenseAV GitHub implementation without any modifications. All results related to DenseAV reported in our paper were obtained using the official DenseAV checkpoint. The differences between our reported results and those presented in the original DenseAV paper can be attributed to two main factors. (1) DenseAV proposes three settings: Sound-only, Speech-only, and Sound-and-Speech. Since the DenseAV paper suggests that a single model can distinguish both sound and speech, one might expect that all reported results use the Sound-and-Speech setting. However, most of the results in the original paper are based on the Sound-only and Speech-only settings, with the exception of the disentanglement evaluation. In contrast, our work focuses on simultaneous grounding of sound and speech within a single model, and thus we adopt the Sound-and-Speech setting as our baseline. (2) The evaluation sample list for AudioSet used in the cross-modal retrieval task was not publicly available. To ensure fairness, we evaluated multiple random combinations of 1,000 AudioSet test samples. While the exact results could not be reproduced, we observed that performance was generally consistent across different splits. We therefore report results from one representative combination. As a result of these differences in evaluation setup, the DenseAV scores

reported in our segmentation and retrieval experiments may appear lower than those presented in the original paper. We include this clarification to help avoid potential confusion and to ensure fair and transparent comparison.

13. Implementation Details

13.1. Regularizers

We incorporate several regularization terms proposed by Hamilton et al. [9] to improve training stability. We re-emphasize that while these techniques do not significantly impact the model's performance, they contribute to more stable training. **Disentanglement Regularizer** encourages different similarity volumes to specialize in distinct audio-visual associations by penalizing simultaneous activations across heads:

$$\mathcal{L}_{DisReg} = \text{Mean}(|S(\mathbf{a}_b, \mathbf{v}_b)[1] \circ S(\mathbf{a}_b, \mathbf{v}_b)[2]|), \quad (13)$$

where $\circ$ denotes element-wise multiplication and $\mathbf{a}_b$, and $\mathbf{v}_b$ refers to audio and image feature of b^{th} sample from a batch respectively. The **Stability Regularizer** consists of several smaller regularization terms. The **Negative Audio Splicing Regularizer** prevents self-attention mechanisms from collapsing by relying exclusively on specific tokens. It introduces negative audio regions into audio clips and penalizes activations in these regions. This is defined as:

$$\mathcal{L}_{Splice} = \text{WeightedMean}(S(\mathbf{a}_b, \mathbf{v}_b)^2, m_b), \quad (14)$$

where m_b represents a soft mask identifying spliced regions. The **Calibration Regularizer** ensures that the calibration temperature τ remains stable by penalizing values over 1.0, expressed as:

$$\mathcal{L}_{Cal} = \max(\log(\tau), 0)^2. \quad (15)$$

The **Non-Negative Pressure Regularizer** promotes positive feature similarity by penalizing similarity scores below zero:

$$\mathcal{L}_{NonNeg} = \frac{1}{|\Omega|} \sum_{\Omega} \min(S(\mathbf{a}_b, \mathbf{v}_{b'})[k, f, t, h, w], 0)^2, \quad (16)$$

where Ω is a set of randomly selected coordinates from the similarity volumes and b' refers to another sample from the batch. Lastly, the **Total Variation Smoothness Regularizer** ensures temporal consistency by penalizing rapid changes in activations over time, defined as:

$$\mathcal{L}_{TV} = \text{Mean}((\text{act}(1:t-1) - \text{act}(2:t))^2), \quad (17)$$

where activations over time are defined as $\text{act}(1:t-1) = (S(\mathbf{a}_b, \mathbf{v}_b)[:, :, t', :, :])_{t'=1}^{t-1}$. Combining these terms, Hamilton et al. [9] defined the overall stability regularizer as:

$$\begin{aligned} \mathcal{L}_{Stability} =& \lambda_{Splice}\mathcal{L}_{Splice} + \lambda_{Cal}\mathcal{L}_{Cal} \\ &+ \lambda_{NonNeg}\mathcal{L}_{NonNeg} + \lambda_{TV}\mathcal{L}_{TV}, \end{aligned} \quad (18)$$

where $\lambda_{Splice} = 0.01$, $\lambda_{Cal} = 0.1$, $\lambda_{NonNeg} = 0.01$, and $\lambda_{TV} = 0.01$.

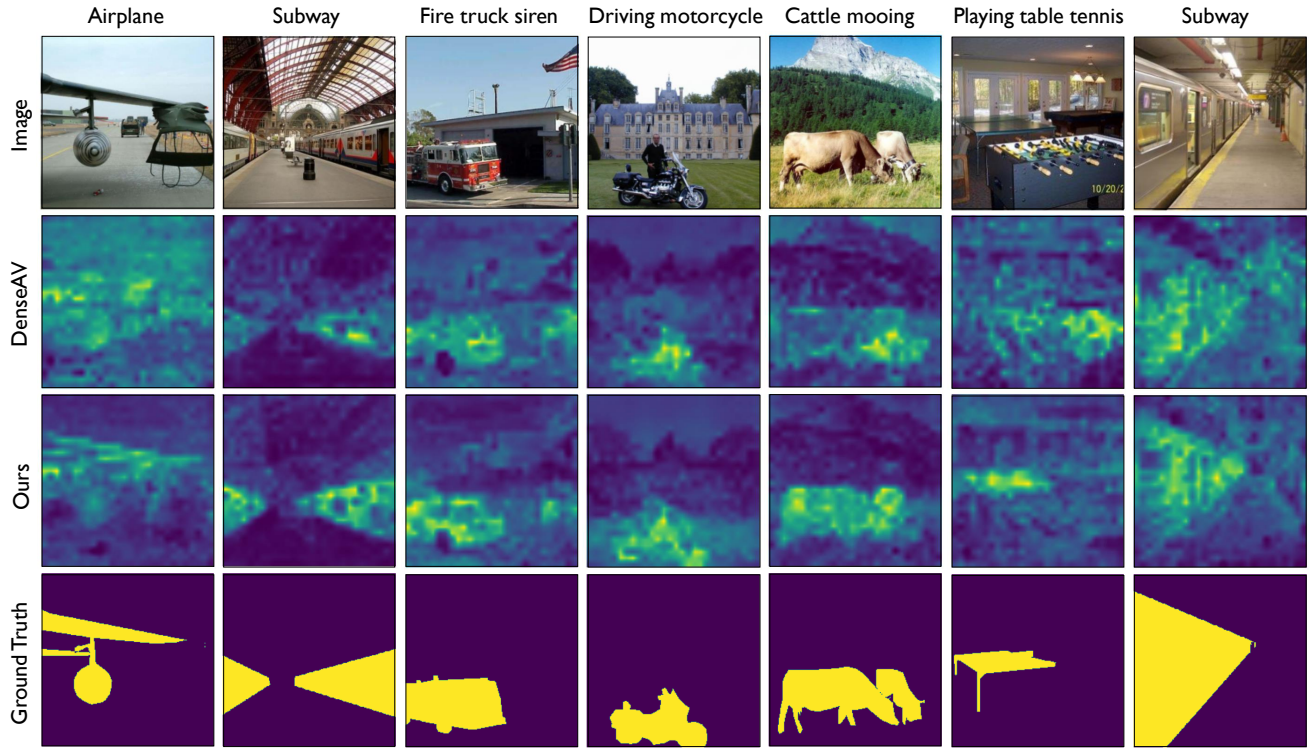

Figure 4. **Sound prompted semantic segmentation on dataset from [9].**

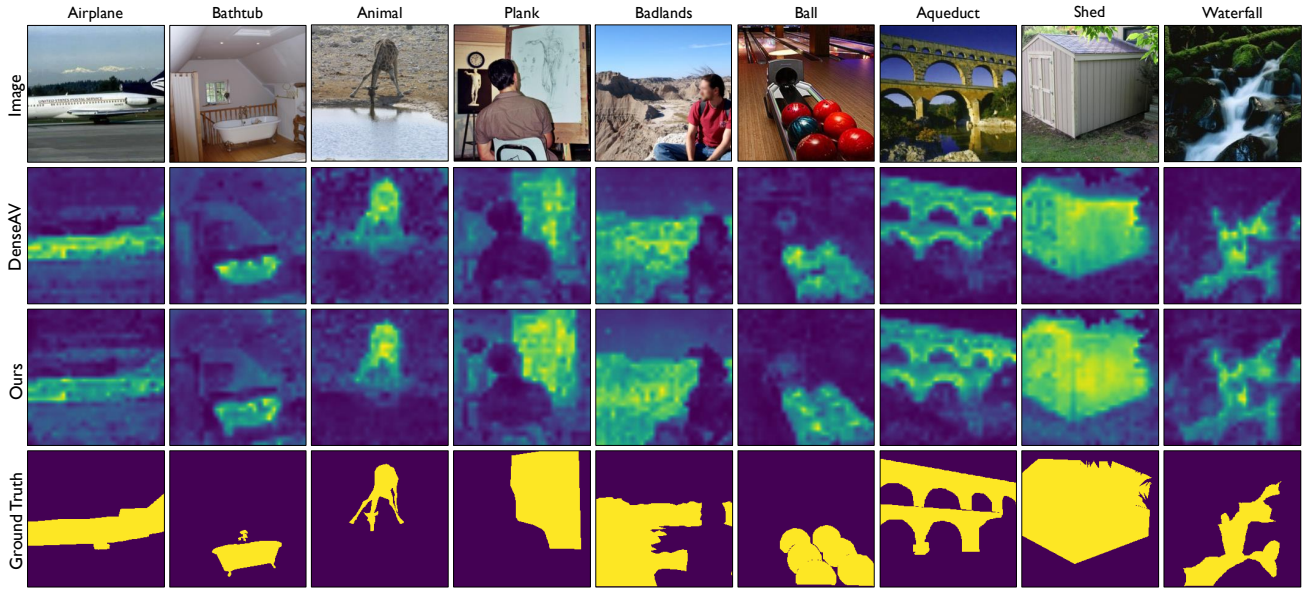

Figure 5. **Speech prompted semantic segmentation on dataset from [9].**

Figure 6. **Simultaneous semantic segmentation on Extended IS3**

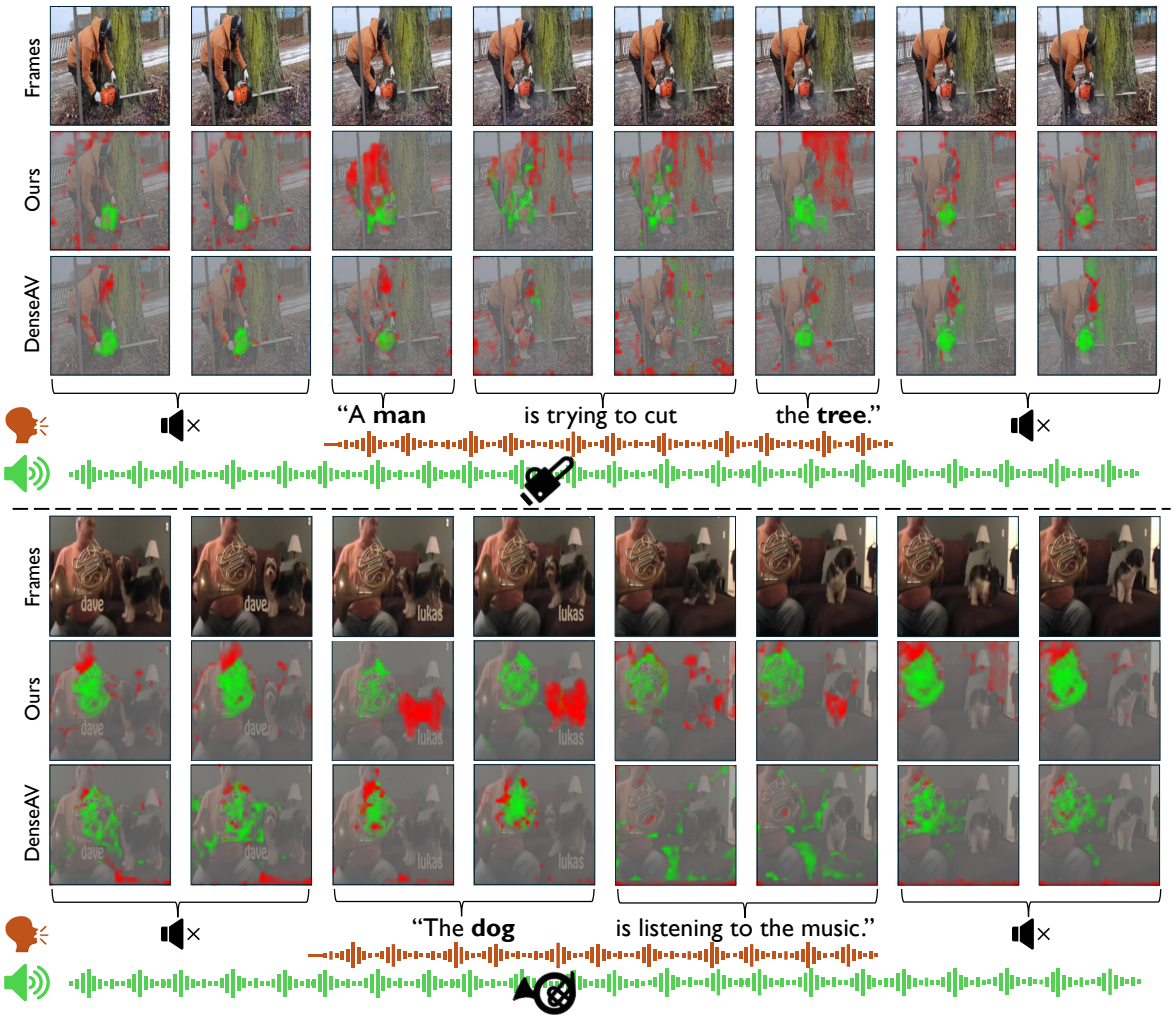

Figure 7. **Real-world scenarios.**

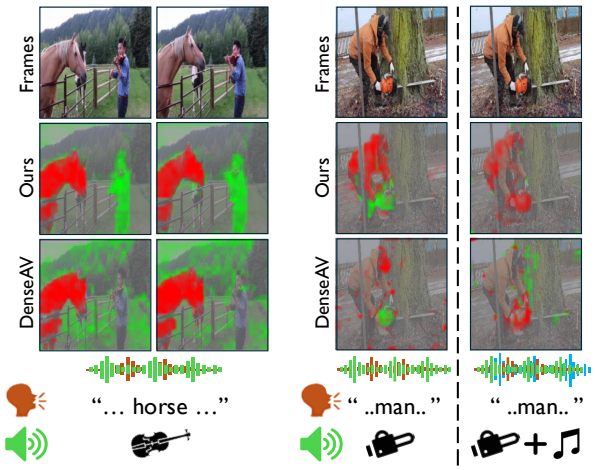

Figure 8. **Failure cases.**