Don't Listen to What You Can't See: The Importance of Negative Examples for Audio-Visual Sound Separation

Efthymios Tzinis[1,2⋆], Scott Wisdom[1], and John R. Hershey[1]

[1] Google Research, Cambridge, MA, USA
[2] University of Illinois Urbana-Champaign, IL, USA

Abstract. For the task of audio-visual on-screen sound separation, we illustrate the importance of using evaluation sets that include not only positive examples (videos with on-screen sounds), but also negative examples (videos that only contain off-screen sounds). Given an evaluation set that includes such examples, we provide metrics and a calibration procedure to allow fair comparison of different models with a single metric, which is analogous to calibrating binary classifiers to achieve a desired false alarm rate. In addition, we propose a method of probing on-screen sound separation models by masking objects in input video frames. Using this method, we probe the sensitivity of our recently-proposed AudioScopeV2 model, and discover that its robustness to removing on-screen sound objects is improved by providing supervised examples in training.

Keywords: Audio-visual, on-screen sound separation, calibration

1 Introduction

Audio-visual on-screen sound separation is the task of recovering only the sounds that originate from on-screen objects in a video. Previously-proposed models extract relevant motion or semantic features from the input visual cues assuming on-screen sounding objects, and have been applied to speech [3,1,7], music [14,5,13,12,6,15,4,16], and more general object classes [2]. Seminal works in on-screen sound separation proposed models that were somewhat invariant to the types of sources [8,5], but those systems were unable to be trained with real-world videos because they required videos labeled as containing on-screen sources.

The recently proposed AudioScope models [9,10] accomplish universal on-screen sound separation and can be trained on real-world videos. This training is enabled by adopting a model architecture that first separates input audio into M sources $\hat{s}_m$, which are then fed to an audio-visual classifier that predicts a logit ℓ_m for each source, corresponding to the log-probability that the source is on-screen. The final on-screen waveform estimate $\hat{x}^{\text{on}}$ is produced using predicted probabilities $\hat{y}_m = \sigma(\ell_m)$ as soft mixing weights for the estimated sources $\hat{s}_m$:

$$\hat{y}_m = \sigma\Big(\hat{\ell}_m\Big) \in [0,1], \quad \hat{x}^{\text{on}} = \textstyle\sum_{m=1}^{M} \hat{y}_m \hat{s}_m. \tag{1}$$

⋆ Work done during an internship at Google.

AudioScope models can be trained unsupervised on mixtures of real-world videos, since the audio-only separation model leverages unsupervised mixture invariant training (MixIT) [11], and the audio-visual classifier uses multiple-instance losses on predicted MixIT assignments of sources to input mixtures.

Audio-visual separation models are usually evaluated on *positive* examples, which are videos that contain at least one on-screen source. The expected behavior of on-screen separation models like AudioScope is to output silence if an input video contains no sounding objects, which we refer to as a *negative* example. If a model does pass audio from such a video, this is analogous to a "false positive" for a binary classifier. Thus, we argue it is important to include such negative examples in evaluation datasets for audio-visual separation models. In this paper, we use AudioScopeV2 as a representative on-screen separation model. We review metrics that measure both reconstruction of on-screen sounds and rejection of off-screen sounds, as well as a calibration procedure to balance these two metrics [10]. We also use negative examples constructed by modifying input video frames to probe the sensitivity of AudioScopeV2.

2 Metrics and Calibration

For on-screen examples with input audio $x = x^{\text{on}} + x^{\text{off}}$, the reconstruction fidelity of an on-screen estimate $\hat{x}^{\text{on}}$ can be measured using *signal-to-noise ratio* (SNR) in decibels (dB) as $\text{SNR}(x^{\text{on}}, \hat{x}^{\text{on}})$. For negative examples with input audio $x = x^{\text{off}}$ where no audio originates from on-screen objects, a model's ability to suppress off-screen sources is measured using *off-screen suppression ratio* (OSR).

$$\text{SNR}(y, \hat{y}) = 20 \log_{10} \frac{\|y\|}{\|y - \hat{y}\|}, \text{OSR}(x, \hat{x}^{\text{on}}) = 20 \log_{10} \frac{\|x\|}{\|\hat{x}^{\text{on}}\|}, \tag{2}$$

where y is a target signal and $\hat{y}$ its estimate. OSR measures the power reduction of the on-screen estimate $\hat{x}^{\text{on}}$ relative to entirely off-screen input audio $x = x^{\text{off}}$.

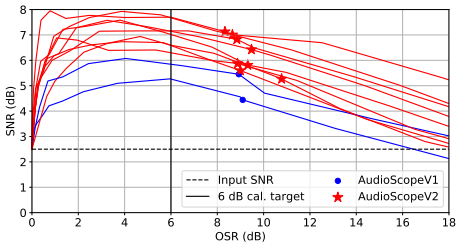


Fig. 1. OSR versus $\text{SNR}(x^{\text{on}}, \hat{x}^{\text{on}})$ curves when calibration offset θ in (3) is varied, on unfiltered random background test set for AudioScopeV1 [9] and AudioScopeV2 [10] models. Points show uncalibrated operating points resulting from training.

Both OSR and SNR are important, but there is an inherent trade-off between them, since OSR can always be increased by scaling down the on-screen estimate $\hat{x}^{\text{on}}$, at the expense of SNR. This makes it difficult to compare models that have different operating points in this trade-off. To illustrate this, Figure 1 plots OSR versus SNR for various AudioScope models. Notice that each model achieves a different operating point. Because of these differences, SNR cannot be meaningfully compared across models without considering OSR. For example, one of the AudioScopeV2 models (red) achieves a lower SNR of about 5.3

dB at 10.8 dB OSR, compared to an AudioScopeV1 model (blue) that achieves a higher SNR of 5.5 dB at 8.9 dB OSR. It is difficult to say which model is better.

Unfortunately, it is difficult to enforce a desired operating point during training. To solve this problem, we can use a post-training calibration method [10] where a classifier bias is adjusted to achieve on average a target OSR. This is akin to choosing a threshold for a detector to achieve a target false positive rate. For our procedure, we define a calibrated on-screen estimate given by adding a global scalar offset θ to the on-screen logits $\hat{\ell}_{1:M}$ (1):

$$\tilde{x}^{\text{on}}(\theta) = \textstyle\sum_m \hat{s}_m \sigma(\hat{\ell}_m + \theta). \tag{3}$$

The offset θ is tuned such that the median OSR (2) across all N_{off} negative examples, $\text{med}_{j=1}^{N_{\text{off}}} \text{OSR}\left[x_j, \tilde{x}_j^{\text{on}}(\theta)\right]$, is equal to a desired $\text{OSR}_{\text{target}}$. The curves in Figure 1 illustrate the effect on OSR and SNR when θ is varied, and are akin to receiver operating characteristic or precision-recall curves.

In practice, θ has a monotonic relationship to OSR. As $\theta \to 0$ (inversely ∞), the on-screen probabilities $\hat{y}_m \to 0$ (inversely 1), and thus OSR approaches ∞ (inversely 0) dB. Because of this property, optimization of θ is very simple, and can be accomplished efficiently via binary search.

In [10], we chose a target OSR of 6 dB. The calibrated operating points are at the intersections of the blue and red lines with the solid line at 6 dB OSR in Figure 1. Notice that calibration resolves the ambiguity between model operating points, allowing comparison of models with a single metric, SNR. This comparison shows that the AudioScopeV2 models (red) are clearly better than AudioScopeV1 models (blue) in a region close to 6 dB median target OSR.

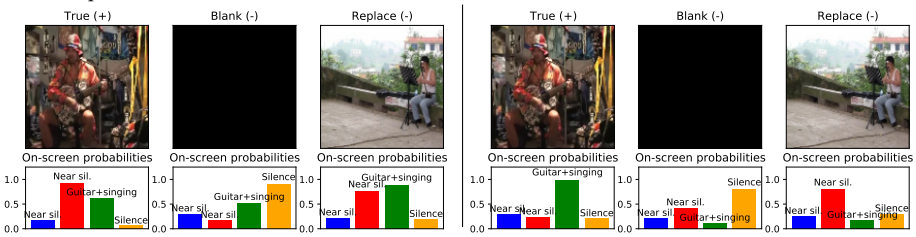
3 Experiment with Negatives

To probe the robustness of AudioScopeV2 to negative examples, we use human-labeled on-screen only videos and construct negative examples using either blank video frames (i.e. all zeros), or replace video frames from another random on-screen-only video. These negative examples test whether the models are properly leveraging visual information to inform the estimate of on-screen sound.

One can consider these negatives as off-screen-only examples, thus we measure OSR (2) on them. In addition, to gauge how much of the original audio is reconstructed, we also measure $\text{SNR}(x, \hat{x}^{\text{on}})$ (2) between the original input audio x and the on-screen estimate $\hat{x}^{\text{on}}$. For positive examples, this is equivalent to $\text{SNR}(x^{\text{on}}, \hat{x}^{\text{on}})$, whereas for negative examples, it is equivalent to $\text{SNR}(x^{\text{off}}, \hat{x}^{\text{on}})$. Table 1 shows these results for unsupervised [10, §3.5] and semi-supervised [10, supplementary §3.3] separable self-attention AudioScopeV2 models. For positive examples where the true video frames are used, the best SNR we can expect is ∞ dB, which indicates perfect recovery of on-screen audio. For negative examples, the best SNR and OSR are 0 dB and ∞ dB, respectively, because the model should produce silence for the on-screen estimate. Models are calibrated to 6 dB OSR on the test set of [10], at which they achieve 6.6 dB SNR for unsupervised and 7.0 dB SNR for semi-supervised.

Table 1. SNR and OSR for calibrated unsupervised and semi-supervised models on positive (+) or negative (-) examples created using on-screen-only videos. SNR is $\text{SNR}(x, \hat{x}^{\text{on}})$, where x is the same input audio across (+) and (-). "Ideal" row provides the ideal metric values, and up/down arrows indicate desired metrics behavior.

	True video (+)	**Blank video (-)**		**Replaced video (-)**	
Training	**SNR (↑)**	**SNR (↓)**	**OSR (↑)**	**SNR (↓)**	**OSR (↑)**
Ideal	∞	0.0	∞	0.0	∞
Unsupervised	**12.6**	6.7	4.9	3.0	9.2
Semi-supervised	8.3	**1.3**	**16.8**	**1.8**	**14.2**



Video frames from "sUPEREDS vote for me song" by sUPEREDbodyforhire, CC-BY-SA 2.0 (left), "MVI_6134" by kenner116, CC-BY 2.0 (right)

Fig. 2. Predicted on-screen probabilities $\hat{y}_m$ for unsupervised (left) and semi-supervised (right) models for a particular on-screen-only example. Separated sources are predicted by the audio-only separation stage, so are the same for all plots, and only one non-zero source is predicted (guitar+singing, green bar). Source labels are annotated manually.

On the test set from [10] composed of on-screen-only videos plus additional off-screen audio background, the semi-supervised model achieves slightly better SNR at 6 dB OSR (7.0 dB > 6.6 dB). However, on the isolated on-screen-only videos considered in Table 1, unsupervised models achieve significantly better SNR on the positive "true video" examples compared to semi-supervised models (12.6 dB > 8.3 dB). However, only looking at these positives does not tell the complete story. In terms of OSR on the negative examples, the unsupervised models have poor rejection of off-screen sounds (lower OSRs of 4.9 dB and 9.2 dB), and still attempt to reconstruct the input audio to some degree (SNRs of 6.7 dB and 3.0 dB). On the other hand, semi-supervised models achieve much better performance on negative examples, achieving OSRs of 16.8 dB and 14.2 dB. These results show that semi-supervised models are more robust to negative examples, and demonstrates the importance of using negative examples in addition to positive examples to gauge the performance of audio-visual models.

Figure 2 shows the video frames and predicted probabilities for a particular example, for the unsupervised and semi-supervised models. Notice that, as Table 1 shows, the semi-supervised model is more robust to negative examples where the input video frames are mismatched from the on-screen audio content.

4 Conclusion

We hope that our proposed metrics and calibration procedure can be useful for other practitioners working on audio-visual models for separation and localization, and that including negative examples in the evaluation datasets for these tasks can yield improved robustness for such models.

References

1. Afouras, T., Chung, J.S., Zisserman, A.: The Conversation: Deep audio-visual speech enhancement. Proc. Interspeech pp. 3244–3248 (2018)
2. Chatterjee, M., Le Roux, J., Ahuja, N., Cherian, A.: Visual scene graphs for audio source separation. In: Proc. IEEE International Conference on Computer Vision (ICCV). pp. 1204–1213 (2021)
3. Ephrat, A., Mosseri, I., Lang, O., Dekel, T., Wilson, K., Hassidim, A., Freeman, W.T., Rubinstein, M.: Looking to listen at the cocktail party: a speaker-independent audio-visual model for speech separation. ACM TOG **37**(4), 1–11 (2018)
4. Gan, C., Huang, D., Zhao, H., Tenenbaum, J.B., Torralba, A.: Music gesture for visual sound separation. In: Proc. IEEE International Conference on Computer Vision (CVPR). pp. 10478–10487 (2020)
5. 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)
6. Gao, R., Grauman, K.: Co-separating sounds of visual objects. In: Proc. IEEE International Conference on Computer Vision (CVPR). pp. 3879–3888 (2019)
7. Gao, R., Grauman, K.: VisualVoice: Audio-visual speech separation with cross-modal consistency. In: Proc. IEEE International Conference on Computer Vision (CVPR). pp. 15490–15500 (2021)
8. Owens, A., Efros, A.A.: Audio-visual scene analysis with self-supervised multisensory features. In: Proc. European Conference on Computer Vision (ECCV). pp. 631–648 (2018)
9. Tzinis, E., Wisdom, S., Jansen, A., Hershey, S., Remez, T., Ellis, D.P., Hershey, J.R.: Into the wild with audioscope: Unsupervised audio-visual separation of on-screen sounds. In: Proc. International Conference on Learning Representations (ICLR) (2021)
10. Tzinis, E., Wisdom, S., Remez, T., Hershey, J.R.: AudioScopeV2: Audio-visual attention architectures for calibrated open-domain on-screen sound separation. In: Proc. European Conference on Computer Vision (ECCV) (2022)
11. Wisdom, S., Tzinis, E., Erdogan, H., Weiss, R.J., Wilson, K., Hershey, J.R.: Unsupervised sound separation using mixtures of mixtures. In: Advances in Neural Information Processing Systems. vol. 33, pp. 3846–3857 (2020)
12. Xu, X., Dai, B., Lin, D.: Recursive visual sound separation using minus-plus net. In: Proc. IEEE International Conference on Computer Vision (CVPR). pp. 882–891 (2019)
13. Zhao, H., Gan, C., Ma, W.C., Torralba, A.: The sound of motions. In: Proc. IEEE International Conference on Computer Vision (CVPR). pp. 1735–1744 (2019)
14. Zhao, H., Gan, C., Rouditchenko, A., Vondrick, C., McDermott, J., Torralba, A.: The sound of pixels. In: Proc. European Conference on Computer Vision (ECCV). pp. 570–586 (2018)
15. Zhu, L., Rahtu, E.: Separating sounds from a single image. arXiv preprint arXiv:2007.07984 (2020)
16. Zhu, L., Rahtu, E.: Visually guided sound source separation and localization using self-supervised motion representations. In: Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 1289–1299 (2022)