Telling Left from Right: Learning Spatial Correspondence of Sight and Sound (Supplementary Material)

Karren Yang*
MIT

Bryan Russell
Adobe Research

Justin Salamon
Adobe Research

http://karreny.github.io/telling-left-from-right

The Supplemental includes video examples corresponding to Figure 1 of the main text as well as additional details of dataset collection and experiments. Section A describes our collection and curation of the YouTube-ASMR-300K and YouTube-ASMR datasets. Sections B, C and D provide additional details of implementation and baselines for downstream tasks. Finally, Section E describes the implementation details and results of extending the audio-visual spatial correspondence task to 360-degree videos.

A. YouTube-ASMR Dataset

Collecting YouTube URLs. We obtained an initial set of approximately 2K URLs of YouTube videos related to ASMR using the search keywords such as "ASMR Binaural", "ASMR Ear-to-ear", "ASMR Tingles" and "ASMR Tapping". Subsequently, we expanded this list by scraping the URLs of related videos in multiple iterations, obtaining a final list of approximately 80K unique URLs. The average length of each video in this list was approximately 30 minutes long based on metadata information. For each URL, we downloaded up to 10-15 video clips of 10 s duration, for a total initial count of approximately 1M video clips.

Data filtering. We filtered the downloaded video clips based on separate criteria for the visual and the audio streams. For the audio stream, we filtered for videos with stereo sound that was perceived to be off-center. In particular, we computed the log difference in the magnitude spectrogram between the left and right stereo channels and produced a weighted average of the difference over the frequency domain based on the amplitude, as proposed by [11]. We then filtered for video clips in which at least 60% of these difference values over time bins was significantly different from zero. For the visual stream, we performed face detection [2] and filtered for video clips that contained a face with high confidence. We also removed videos containing still images by computing differences between frames over different time steps. Following these data filtering steps, the resulting dataset, YouTube-ASMR-300K, consists of over 300K video clips from approximately 30K unique YouTube URLs.

*Work done at Adobe Research during KY's summer internship.

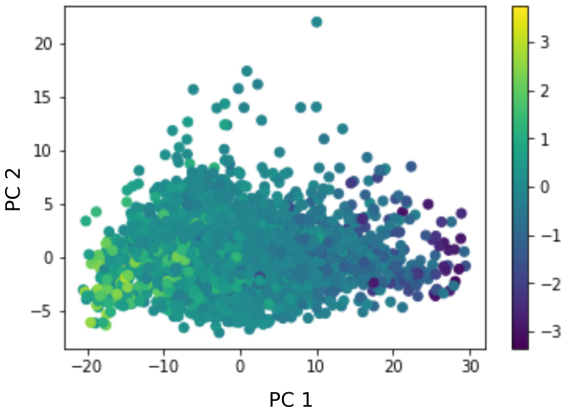

Figure 1. PCA visualization of audio embeddings extracted from YouTube-ASMR dataset, colored by log energy-difference between left and right audio channels. Each point corresponds to a different video in the held-out data.

Manual curation of YouTube-ASMR subset. For additional quality control, we selected 50 top YouTube channels from the YouTube-ASMR-300K dataset and manually assessed one-third of their videos. We then created a subset of the YouTube-ASMR-300K dataset comprising of the videos from 30 YouTube channels with strong audio-visual spatial correspondence based on our manual inspection.

B. Sound Localization

Additional details of correlation analysis. Here we provide additional information corresponding to the analysis from the beginning of Section 5.1. Recall that we first evaluate whether the audio embeddings learned by our model contain spatial information. We do this by comparing the learned audio features to a sound source's spatial location at each time instance. To obtain an approximate location of a sound source at a time instance, we compute the log-energy difference between the two audio channels. Supplemental

Table 1 shows Pearson R and Spearman R correlation coefficients to measure the correlation between the approximate sound location computed from directional energy and projections of the self-supervised learned embedding. We find that the audio features are strongly correlated with the location of the sound source based on both an unsupervised projection (*i.e.*, principal component analysis) and a supervised projection (*i.e.*, canonical correlation analysis) of the learned embeddings. Additionally, Supplemental Figure 1 shows a PCA plot of the audio embeddings colored by the log energy-difference; observe that the first principal component of the embeddings strongly reflects the log energy-difference.

Next, recall that we investigated whether the visual sub-network has learned to identify the positions of sound sources, as a result of having to match them with the spatial cues provided by the audio sub-network. To determine the regions that the visual sub-network attends to, we use a sliding window of 32 pixels and replace regions of the visual frame with their mean values. We then pass the modified frames through our visual sub-network and determine those regions whose omission maximally affects prediction on the pretext task [4, 12]. Supplemental Table 1 shows our evaluations of the predicted regions of importance using the masking approach of [4, 12] ("Predicted ROI") by comparing them to the approximate ground truth locations based on the log energy-difference between audio channels. We find that the predicted localization results are significantly correlated with the approximate ground truth locations.

Baselines. To preclude the possibility that the visual network is relying on artist-dependent spatial biases to achieve this correlation rather than the dynamic spatial audio cues, we generate baseline predictions of sound localization based on the prior distribution of the sound, visual salience, and motion in training videos produced by the same artists using metadata information. All of these baselines exploit biases in the patterns of sound source localization that may be present in different ASMR artists' videos. The observation that our result outperforms these baselines indicates that the sound source localization is being performed based on audio-visual spatial correspondence, rather than these biases.

- **Sound Prior.** For a given test video clip, we consider all training set video clips made by the same YouTube channel. We then take the directional energy of the sound, *i.e.*, the log energy-difference between the left and right audio channels, averaged over these videos.

- **Visual Salience Prior.** For a given test video clip, we consider all training set video clips made by the same YouTube channel. For each video, we use a horizontal sliding window of 32 pixels to determine the region of the videos with the largest squared distance to the average pixel value of the video. The prediction is given by the mode over these training clips.

Feature	Pearson R	Spearman R
Sound prior	0.062	0.101
PCA proj	0.709	0.790
CCA proj	**0.909**	**0.876**
Salience prior	0.008	0.012
Motion prior	0.009	0.001
Predicted ROI [12]	**0.259**	**0.286**

Table 1. Quantitative evaluation of different sound localization predictions based on Pearson (linear) correlation and Spearman (rank-based) correlation with the log energy-difference between left and right audio channels. Higher is better. The top results are for audio approaches and the bottom ones are for vision. The visual attention of our model outperforms the baselines.

- **Motion Prior.** For a given test video clip, we consider all training set video clips made by the same YouTube channel. For each video, we use a horizontal sliding window of 32 pixels to determine the region of the videos with the most motion, *i.e.*, largest squared distance between subsequent time frames. The prediction is given by the mode over these training clips.

As shown in Supplemental Table 1, these baselines fall short of the performance of the predicted ROI. Overall, our results strongly suggest that the flipping task trains the model to match the locations of sound sources in the visual frames to spatial audio cues, thus learning audio-visual spatial correspondence.

Implementation Details for Sounding Face Tracking. The architecture is shown in Figure 3(b) in the main text. The audio and visual sub-networks have the same architecture as those used for pretext task. The features from the sub-networks are stacked and passed through two deconvolution modules to expand the spatial grid size from 1-by-1 to 7-by-7, followed by convolution modules to ensure 40 outputs per spatial grid cell (4 coordinates and one confidence prediction for each of 8 anchor boxes). We obtain the anchor (prior) boxes using k-means clustering on the training data. We train our models on the YouTube-ASMR dataset, using 1-second audio clips sampled from full clips and a random visual frame outside of this range. The visual frame is resized to 256 x 256 and we shift the color and contrast as a form of data augmentation. For the audio input, we use the stacked log-transformed mel-spectrograms (number of frequency bins=512, window size=400 samples, hop size=160 samples) of the left and right audio channels sampled at 16 kHz computed using Librosa [7]. For optimization, we use Adam [5] with a learning rate of 1e-3, training on approximately 2M samples.

C. Audio Spatialization (Upmixing)

Implementation Details. Our implementation for this task is based on [4]. The architecture is the U-Net shown in Figure 3(c) in the main text. The main difference is that we use multiple video frames to upmix a single audio clip, and we integrate these multiple frames into the U-Net by fusing the visual features with the reduced audio representation along the time dimension. We train and evaluate our models on both our YouTube-ASMR dataset and the FAIR-Play dataset. For both datasets, we use 2.87-second clips sampled from full clips. We use visual frames sampled at 6 Hz with frames resized to 256 x 256, and randomly crop and shift the color/contrast for data augmentation. For the audio input, we use the complex spectrogram (number of frequency bins=512, window size=400 samples, hop size=160 samples) of the difference between the left and right audio channels sampled at 16 kHz [7]. For optimization, we use Adam [5] with a learning rate of 1e-3, training on approximately 2M samples for YouTube-ASMR and 150K samples for FAIR-Play.

Baselines. All of the baselines use the same visual sub-network model architecture (ResNet-18) as our pretrained visual sub-network. For the audio-visual correspondence task baselines, we train models on the tasks described in previous work (*i.e.*, [1] for audio-visual mismatch task and [9, 6] for the temporal shift task) over the same number of samples as our spatial correspondence task. The models have the same architecture as shown in Figure 3(a) in the main text, except they use average pooling rather than spatial flattening in the fusion of the video stream with the audio stream. For the supervised baseline, we use a ResNet-18 model pretrained on ImageNet classification [10].

D. Audio-Visual Source Separation

Implementation Details. Our implementation for this task is based on [4]. We use the same model as the U-Net for audio spatialization, adapted for source separation (as shown in Figure 3(d) of the main text). We train and evaluate our models on both our YouTube-ASMR dataset and the FAIR-Play dataset. For both datasets, we use 2.87-second clips sampled from full clips. We use visual frames sampled at 6 Hz with frames resized to 256 x 256, and randomly crop and shift the color/contrast for data augmentation. For the audio input, we use the stacked magnitude spectrograms (number of frequency bins=512, window size=400 samples, hop size=160 samples) of the left and right audio channels sampled at 16 kHz [7]. For optimization, we use Adam [5] with a learning rate of 1e-3, training on approximately 2M samples for YouTube-ASMR and 200K samples for FAIR-Play.

Baselines. For the audio-visual correspondence task baselines, we train models on the tasks described in previous work (*i.e.*, [1] for audio-visual mismatch task and [9, 6] for the temporal shift task) over the same number of samples as our spatial correspondence task. The models have the same architecture as shown in Figure 3(a) in the main text, except they use average pooling rather than spatial flattening in the fusion of the video stream with the audio stream. Similar to the features for our audio-visual correspondence task, we use the joint audio-visual features after the fusion for the source separation task. For the supervised baseline, we use features from a ResNet-18 model pretrained on ImageNet classification [10].

E. Spatial Alignment in 360-Degree Video

Here we discuss the implementation details and results of the audio-visual spatial correspondence task in 360-degree videos.

Implementation Details. We generally use the same model structure as the one we used for field-of-view video and stereo audio, as depicted in Figure 3(a) of the main text: the model consists of distinct visual and audio sub-networks that are fused prior to classification, and we reduce and flatten the visual features prior to fusion with the audio. The main difference is that the audio input is four channels instead of two. We train our model on the YouTube-360 dataset consisting of 360-degree videos with first-order ambisonics (FOA) audio [8]. The dataset consists of over a thousand 360-degree videos with corresponding spatial audio, containing a variety of content including street views and musical performances. We use 3 second video clips sampled from full-length videos, introducing negative (rotated audio) examples with probability 0.5 with $\theta \in [0.95\pi, 1.05\pi]$. We use 360-degree videos sampled at 5 Hz with frames (as equirectangular projection) resized to 240 x 480. To augment the dataset, we randomly shift the color/contrast of the videos and apply random rotation about the z-axis of both video (via translation of the equirectangular projection) and spatial audio. For the W-channel, we use the short-time Fourier transform (STFT) of audio sampled at 16 kHz, and for the other channels, we take the inner product of the channel STFT with the W-channel STFT, which is an effective input for spatial audio localization tasks using neural networks [3].

Results. Supplemental Table 2 shows the test classification accuracy of the spatial alignment task trained on YouTube-360 dataset. As a baseline, we initialized our visual sub-network using a model trained on ImageNet classification. We found that our model with visual sub-network trained from scratch performed comparably to this supervised baseline. Compared to earlier spatial alignment on the YouTube-ASMR and FAIR-Play datasets, these accuracy values appear low. We hypothesize that this is due to the in-the-wild nature of the dataset, resulting in sparser signals.

On the other hand, as shown in Supplemental Table 2,

Model	Test Accuracy
ResNet-18 (Supervised)	0.57791
No W-Channel	0.61395
No XYZ-Channels	0.49535
ResNet-18	0.58605

Table 2. Performance on pretext task: YouTube-360 dataset. The models perform comparably except the XYZ-channel ablation that eliminates spatial information.

	Alignment Error (Degrees)		
Model	Low Conf	Med Conf	High Conf
Original	65.34	50.51	19.06
Shorter clips	71.05	55.23	38.21
Harder negatives	74.69	66.65	47.00

Table 3. Evaluation of pretext models on audio-visual alignment in 360-degree videos, in degrees of error. Worst is 180 degrees, random is 90 degrees. Low conf shows the result averaged over all videos. Medium conf and high conf refer to approximately the 50th and 90th percentiles of videos based on the confidence of the model's classification, *i.e.*, predicted probability of a video being aligned correctly.

we find that a model with only 60% classification accuracy on 3-second clips aligns 360-degree videos with high accuracy when evaluated over the entire video (approximately 19 degrees of rotational error for high confidence videos). Specifically, for each video in the YouTube-360 test dataset, we compute the output of our model for uniform rotations of the spatial audio around the z-axis over multiple sub-sampled clips per video. We use consensus scoring to generate probabilities for different rotation angles and then compute the absolute value of the alignment error weighted by these probabilities. The results, specifically prediction errors in degrees, are shown in Table 3. As ablations/comparisons, we also trained the pretext task on shorter clips (1 s) and harder negatives with smaller rotation angle of misaligned examples ($\theta \in [0.25\pi, 1.75\pi]$). We found that making the pretext detection task more challenging (*i.e.*, decreasing the clip length or increasing the difficulty of the negative examples) did not improve performance on this task.

Qualitative analysis. Qualitatively, we observe that the features learned by the pretrained audio sub-network capture the azimuth angle of the sound localization. Supplemental Figure 2 visualizes the first two principal components of audio embeddings extracted from the Tau Spatial Sound dataset, colored by the azimuth angle label. Notice that the angle of the first two principal components reflects the azimuth angle of the sound's direction of arrival. This observation enables us to perform acoustic localization using the learned audio embedding, *i.e.*, by overlaying the angle that is estimated based on the first two principal components over the equirectangular projection of the 360-degree video.

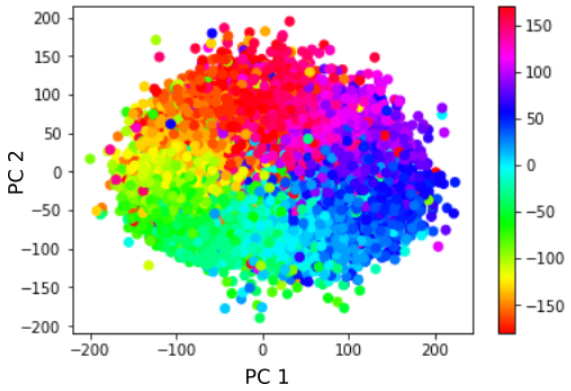

Figure 2. PCA visualization of audio embeddings extracted from Tau Spatial Sound dataset, colored by azimuth angle (in degrees). Each point corresponds to a different audio clip from the dataset.

References

[1] Relja Arandjelovi and Andrew Zisserman. Look, listen and learn. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017. 3

[2] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 1

[3] Yin Cao, Turab Iqbal, Qiuqiang Kong, Miguel Galindo, Wenwu Wang, and Mark Plumbley. Two-stage sound event localization and detection using intensity vector and generalized cross-correlation. Technical report, DCASE2019 Challenge, June 2019. 3

[4] Ruohan Gao and Kristen Grauman. 2.5D visual sound. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2, 3

[6] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 3

[7] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, 2015. 2, 3

[8] Pedro Morgado, Nuno Vasconcelos, U C San, Diego Timothy Langlois, and Oliver Wang. Self-supervised generation of spatial audio for 360° video. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 3

[9] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018. 3

[10] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017. 3

[11] M Vinyes, Jordi Bonada, and Alex Loscos. Demixing commercial music productions via human-assisted time-frequency masking. In *Proceedings of Audio Engineering Society 120th Convention*, 2006. 1

[12] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 2