Active Audio-Visual Separation of Dynamic Sound Sources Supplementary Material

Sagnik Majumder[1] and Kristen Grauman[1,2]

[1] UT Austin, Austin, TX, USA
[2] Facebook AI Research, Austin, TX, USA
{sagnik, grauman}@cs.utexas.edu

In this supplementary material we provide additional details about:

- Video (with audio) for qualitative depiction of our task and qualitative evaluation of our agent's performance (Sec. 1).
- Failure cases of our method (Sec. 2), as referenced in Sec. 5.2 in the main paper.
- Separation quality of our method as a function of agent location in the 3D environment (Sec. 3), as referenced in Sec. 5.2 in the main paper.
- Experiment to analyze the effect of our multi-step prediction parameter E (Sec. 4.1 in main) on the dynamic separation quality (Sec. 4), as mentioned in Sec. 5 of the main paper.
- Experiment to gauge robustness of our method to audio noise (Sec. 5), as referenced in Sec. 5.2 of the main paper.
- Experiment to show the dependence of dynamic separation quality on the number of audio sources (Sec. 6), as noted in Sec. 5.2 of the main paper.
- Experiment to show the dependence of dynamic separation quality on the minimum inter-source distance (Sec. 7), as referenced in Sec. 5.2 of the main paper.
- Audio data details (Sec. 8), as mentioned in Sec. 5 of the main paper.
- Baseline details for reproducibility (Sec. 9), as noted in Sec. 5 of the main paper.
- Model architectures details for our method and baselines (Sec. 10), as noted in Sec. 5 of the main paper.
- Training hyperparameters (Sec. 11), as referenced in Sec. 5 of the main paper.
- Definition of metrics used for measuring separation quality (Sec. 12), as mentioned in Sec. 5 of the main paper.

1 Qualitative Video

The supplementary video, available at `https://vision.cs.utexas.edu/projects/active-av-dynamic-separation/`, demonstrates the SoundSpaces [3] audio simulation platform that we use for our experiments, provides a qualitative depiction of our task, Active Audio-Visual Separation of Dynamic Sound Sources, and also illustrates the technical contributions of our approach over Move2Hear [11], a state-of-the-art method for active

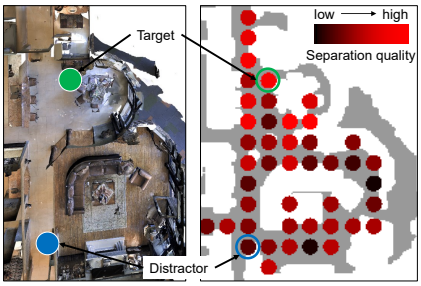

Fig. 1: Separation quality of our model when placed at different locations in a 3D scene for a given target source and distractor.

audio-visual separation of static sounds. Moreover, we qualitatively compare our method with the best-performing heuristical baseline, namely Proximity Prior (Sec. 5 in main), and Move2Hear [11] (Sec. 5 in main), as well as qualitatively analyze failure cases. Please use headphones to hear the spatial audio correctly.

2 Failure Cases.

Our agent fails to separate well when it moves too far away from the target in search of separation-friendly spots and hence cannot track the target after a point, even with the help of its transformer memory. Other failure cases involve the agent being stranded among multiple close-by audio distractors, while having very limited scope of movement due to the surrounding 3D structure. For examples of failure episodes, watch our qualitative video (Sec. 1).

3 Separation Quality vs. Agent Location

We show how our agent's separation quality evolves during its trajectory as a function of the observation poses it chooses, in the form of heatmaps in Fig. 4 in main and Supp video (4:57 - 5:19; 6:57 - 7:19). The fill color of a circle on the trajectory indicates its separation quality. Additionally, if we place our model at fixed locations and measure the separation quality (i.e., non active setting), we can produce a heatmap like Fig. 1 where we can see how separation quality varies as a function of scene geometry.

4 Multi-step prediction parameter E

On setting our multi-step prediction parameter E (Sec. 4.1 in main) to non-zero values other than 14 (Sec. 5 in main), the overall separation quality (SI-SDR) on *unheard* sounds drops up to 6.6%. See Fig. 2. Lower E values reduce our model's ability to improve past separations given the new observations. However,

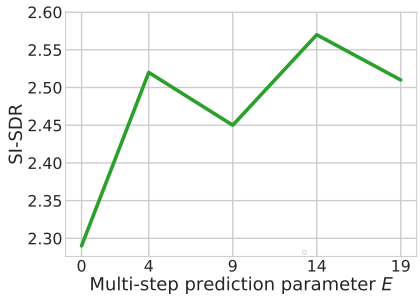

Fig. 2: Effect of our multi-step prediction parameter E on dynamic separation quality. Higher SI-SDR is better.

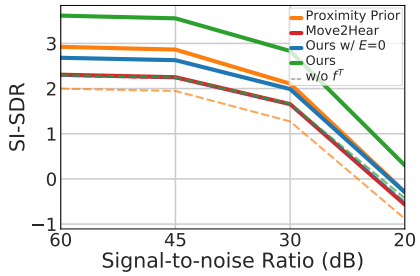

Fig. 3: Models' robustness to various noise levels in audio. Higher SI-SDR is better.

on increasing E beyond a certain level, we see a degradation in separation quality as well. We expect that when the predictions are temporally distant, they are likely to be dissimilar in nature and their quality gets adversely affected due to the lack of useful cues in the current estimate for backward transfer.

5 Audio Noise

We test our method's dynamic separation performance in the presence of standard microphone noise [21,20] (Sec. 5.2 in main). See Fig. 3. Our method robustly holds its superiority in separation over strong baseline models even at high noise levels (e.g., SNR of 20 dB, 30 dB, etc. [3]) See and hear our supplementary video (Sec. 1) to get a better understanding of the distortion caused by the maximum noise that we evaluate, *i.e.*, SNR = 20 dB, where we play some noisy mixed binaural samples, as heard from the agent's initial pose. Our method benefits from having the transformer memory f^T (Sec. 4.1 in main), without which its separation performance drops sharply. Further, our multi-step prediction mechanism (Sec. 4.1 in main) for improving older separations in addition to separating the dynamic audio target for the current step provides a substantial boost in performance across all noise levels (compare the green and blue curves).

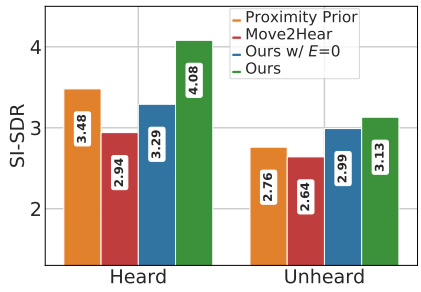

Fig. 4: Dynamic separation quality with 3 sources, *i.e.*, , 2 distractor sounds. Higher SI-SDR is better.

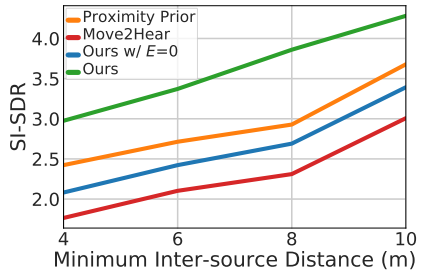

Fig. 5: Dynamic separation quality with minimum inter-source distance. Higher SI-SDR is better.

6 Number of Audio Sources

We evaluate how increasing the number of dynamic distractor sounds in the environment affects separation performance (Sec. 5.2 in main). Fig. 4 shows that even with $k = 3$ sources (*i.e.*, , 1 target and 2 distractors) in every episode, our model leverages its smart motion policy and transformer memory to generalize better than the strongest of baselines. Further, multi-step predictions help tackle more distractors with both *heard* and *unheard* sounds (compare blue and green bars).

7 Minimum Inter-Source Distance

We examine how changing the minimum inter-source distance for every episode in our dynamic audio setting affects the separation performance (Sec. 5.2 in main). See Fig. 5. Our method is able to maintain its performance gain over the most competitive baselines and also its own ablated version that makes single-step predictions. This is true even at very low values of inter-source distance, where our agent is severely cramped for room to move in search of separation-friendly spots, due to the close proximity to distractor sounds. This highlights the robustness to

variations in the relative spatial arrangement of the target and distractor sources, which we can attribute to our method's joint learning of a dynamic separation policy and a transformer memory for making multi-step predictions for past and current targets.

8 Audio Data

Here, we elaborate on the details of the audio data that we use for our experiments (Sec. 5 in main).

Monaural Audio Dataset. Our monaural audio dataset contains 100 speaker classes from LibriSpeech [8], 1 music class of different instruments from the MUSIC [25] dataset, and 1 class of assorted background sounds from ESC-50 [14] (Sec. 5 in main).

For LibriSpeech, each of the 100 speakers has at least 25 minutes of audio data in total. For all speech types and music, we join audio clips from the same type to form long audio clips, each of which is at least 20 s long, before splitting them into non-overlapping clips for train/val/test splits for unheard sounds (Sec. 5 in main). For background sounds from our ESC-50 data, we replicate each 5 s clip four times and join them end to end to produce long clips of 20 s length. We choose replication for ESC-50 because joining distinct clips often doesn't lead to natural sounding audio due to high intra-class variance in the original dataset.

We resample all clips at 16kHz and encode them using the standard 32-bit floating point format. Next, we compute the mean power across all clips and normalize each clip such that its power is equal to the pre-computed mean of 1.44. As a result, the mean (±std) audio amplitude in our dataset is 2.4 (±399.7) for speech, -1.0 (±399.6) for music, and -0.7 (±399.6) for background sounds. The high std values for all sound categories indicate the high levels of dynamicity in all monaural clips, which not only make the dynamic separation task realistic but also challenging in nature.

Having preprocessed the long audio clips, we play a fresh audio segment at every dynamic source during an episode by sampling a random starting point within the long audio clips and shifting it forward by 1 s after every step. We loop over from the clip start if its end is reached during the course of an episode.

Spectrogram Computation. To compute spectrograms, we use the Short-Time Fourier Transform (STFT) with a Hann window of length 63.9ms, hop length of 32ms to promote significant temporal overlap of consecutive windows, and FFT size of 1023. This generates complex spectrograms of size $512 \times 32 \times C$, where C is the number of channels in the source audio (1 for monaural and 2 for binaural). For all experiments, we use the magnitude of a complex spectrogram after reshaping it to $32 \times 32 \times 16C$, taking slices along the frequency dimension and concatenating them channel-wise to make model training computationally tractable. For all modules in our method and the baselines that take spectrograms as an input, except for the monaural audio encoder $\mathcal{F}^M$ (Sec. 4.1 in main) of the

active motion policies, we compute the natural logarithm of the spectrograms by adding 1 to all their elements for better contrast [5,6], before feeding them to the respective modules.

Whenever the type label of the target audio source needs to be fed to a module along with a magnitude spectrogram, its value is looked up in a pre-computed dictionary with type names for keys and positive integers for label values, and concatenated with the input spectrogram after slicing.

9 Baseline Details

We provide additional details about our baselines for reproducibility.

- **DoA:** To face the direction of arrival (DoA) of the target audio, this agent first rotates clockwise from its starting pose until there is an adjacent node in front of it that's connected to the one it is currently at, then moves to the neighboring node along the connecting edge, and finally turns twice in the clockwise direction before holding its pose through the rest of episode for sampling direct sound from the target.
- **Novelty [2]:** this agent is incentivized to maximize its coverage of novel states in its environment. In our setup, each node of the SoundSpace [3] grids is considered to be a unique state, which has an associated visitation count value that starts from 0 and is incremented every time the agent visits that state. At step t, the agent receives a reward:

$$r_t = \frac{1}{\sqrt{n_s}}, \tag{1}$$

  where n_s is the visitation count of its current state s_t.
- **Move2Hear [11]**: to account for dynamic audio, we modify the monaural audio encoder of its active audio-visual controller to only receive $\ddot{M}_t^G$ in place of the channel-wise concatenation of $\tilde{M}_t^G$ and $\ddot{M}_t^G$.

10 Model Architectures

Passive Audio Separator. The passive audio separator f^P comprises a binaural extractor f^B for extracting the target binaural given the mixed audio and a target type, and a monaural converter for predicting the target monaural from the extracted binaural (Sec. 4.1 in main).

f^B and f^M are U-Nets [15] (Sec. 4.1 in main). Their encoder is made of 5 convolution layers, each with a kernel size of 4, a stride of 2, a padding of 1 and a leaky ReLU [13,19] activation with a negative slope of 0.2. The number of convolution output channels are [64, 128, 256, 512, 512], respectively. Their decoder consists of 5 transpose convolution layers, each with a kernel size of 4, a stride of 2, a padding of 1 and a ReLU [13,19] activation. We append a convolution layer with a kernel size of 1 and stride of 1 to the U-Nets to produce the final spectrogram output of the networks.

Transformer Memory. Our transformer memory is a transformer encoder [22] with 2 layers, 8 attention heads, a hidden size of 1024 and ReLU [13,19] activations. It has a pre-norm architecture that's been found to be well-suited for audio separation [18,24]. Instead of using LayerNorm [1] on the additive skip connection output, as proposed in the original transformer design [22], we use LayerNorm [1] on the input side for both the multi-head attention and the feedforward blocks before the additive skip connection branches out. We use this block for every layer of the transformer encoder.

We use a CNN for encoding the current and past monaural estimates $\tilde{M}^G$ from f^P to 1024-dimensional features and add them with the corresponding sinusoidal positional encodings of the same dimensionality (Sec. 4.1 in main), before feeding them to the transformer encoder for self-attention. The encoder has 2 convolutions, each with a kernel size and a stride of 2, and 16 output channels. We use a ReLU activation [13,19] after the first convolution. For decoding the transformer encoder outputs to produce $\ddot{M}^G$s, we use another CNN with 2 transpose convolutions, each with a kernel size and a stride of 2, 16 output channels and a ReLU activation [13,19] (Sec. 4.1 in main). The decoder also receives the output of the first convolutional layer in the encoder as a skip connection, and adds it to the output from its own first transpose convolutional layer, before passing it to the second transpose convolution.

Observation Encoders. The visual encoder $\mathcal{F}^V$ (Sec. 4.2 in main) of our method and all baselines that uses one, namely Novelty [2] and Move2Hear [11], is a CNN with 3 convolution layers with ReLU [13,19] activations, where the kernel sizes are [8, 4, 3], the strides are [4, 2, 1] and the number of output channels are [32, 64, 32], respectively. The convolution layers are followed by 1 fully connected layer with 512 output units.

We use the same architecture as $\mathcal{F}^V$ for $\mathcal{F}^B$ and $\mathcal{F}^M$ (Sec. 4.2 in main), except that we use a kernel size of 2 instead of 3 for the last convolution.

Policy Network. The policy network (Sec. 4.2 in main) for our method and the baselines with RL motion policies (*i.e.*, , Novelty [2] and Move2Hear [11]), comprises a one-layer bidirectional GRU [4] with 512 hidden units, and one fully-connected layer for its actor and critic networks.

We use He-normal [9] weight initialization for all network layers, except for the policy network GRUs, where we use semi-orthogonal weight initialization [17], and the transformer encoder, for which we use the Xavier-uniform [7] initialization strategy.

11 Training Hyperparameters

We pretrain f^P by creating a static dataset of randomly sampled data points (Sec. 4.3 in main), where each scene contributes a maximum of 30K data points, and using the Adam [10] optimizer with an initial learning rate of $5e^{-4}$ and a maximum gradient norm of 0.8 until convergence.

We train the active motion policies of our method, Move2Hear [11] and Novelty [2] for 150 million steps with Decentralized Distributed PPO (DD-PPO) [23], where the weights on the value and entropy loss are 0.5 and 0.01, respectively, and the Adam [10] optimizer with an initial learning of $1e-4$ and a maximum gradient norm of 0.5. We update the policy parameters after every 20 steps of agent's experience for 4 epochs.

To jointly train f^T or the acoustic memory refiner of Move2Hear [11] with the corresponding active motion policy, we use Adam [10] with an initial learning rate of $5e^{-3}$.

12 Separation Quality Metrics

Here, we provide additional details about our metrics for evaluating dynamic separation (Sec. 5 in main).

1. **STFT distance** – The Euclidean distance between the complex spectrograms for the monaural prediction and the ground truth,

$$\mathcal{D}_{\{STFT\}} = ||\ddot{\boldsymbol{M}}^G - \boldsymbol{M}^G||_2.$$

2. **SI-SDR [16]** – We adopt an efficient nussl [12] implementation to compute the scale-invariant source-to-distortion ratio (SI-SDR) of a predicted monaural waveform in dB.

References

1. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
2. Bellemare, M.G., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., Munos, R.: Unifying count-based exploration and intrinsic motivation. arXiv preprint arXiv:1606.01868 (2016)
3. Chen, C., Jain, U., Schissler, C., Gari, S.V.A., Al-Halah, Z., Ithapu, V.K., Robinson, P., Grauman, K.: SoundSpaces: Audio-visual navigation in 3D environments. In: Proceedings of the European Conference on Computer Vision (ECCV). Springer (2020)
4. Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A.C., Bengio, Y.: A recurrent latent variable model for sequential data. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 28. Curran Associates, Inc. (2015), https://proceedings.neurips.cc/paper/2015/file/b618c3210e934362ac261db280128c22-Paper.pdf
5. Gao, R., Grauman, K.: 2.5d visual sound. In: CVPR (2019)
6. Gao, R., Grauman, K.: Co-separating sounds of visual objects. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3879–3888 (2019)
7. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Teh, Y.W., Titterington, M. (eds.) Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 9, pp. 249–256. PMLR, Chia Laguna Resort, Sardinia, Italy (13–15 May 2010), https://proceedings.mlr.press/v9/glorot10a.html
8. Griffin, D., Jae Lim: Signal estimation from modified short-time fourier transform. IEEE Transactions on Acoustics, Speech, and Signal Processing **32**(2), 236–243 (1984). https://doi.org/10.1109/TASSP.1984.1164317
9. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (December 2015)
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Majumder, S., Al-Halah, Z., Grauman, K.: Move2Hear: Active audio-visual source separation. In: ICCV (2021)
12. Manilow, E., Seetharaman, P., Pardo, B.: The northwestern university source separation library. Proceedings of the 19th International Society of Music Information Retrieval Conference (ISMIR 2018), Paris, France, September 23-27 (2018)
13. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: ICML. pp. 807–814 (2010), https://icml.cc/Conferences/2010/papers/432.pdf
14. Piczak, K.J.: ESC: Dataset for Environmental Sound Classification. In: Proceedings of the 23rd Annual ACM Conference on Multimedia. pp. 1015–1018. ACM Press. https://doi.org/10.1145/2733373.2806390, http://dl.acm.org/citation.cfm?doid=2733373.2806390
15. Ronneberger, O., P.Fischer, Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI). LNCS, vol. 9351, pp. 234–241. Springer (2015), http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a, (available on arXiv:1505.04597 [cs.CV])

16. Roux, J.L., Wisdom, S., Erdogan, H., Hershey, J.R.: Sdr – half-baked or well done? In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 626–630 (2019). https://doi.org/10.1109/ICASSP.2019.8683855
17. Saxe, A.M., McClelland, J.L., Ganguli, S.: Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. arXiv preprint arXiv:1312.6120 (2013)
18. Subakan, C., Ravanelli, M., Cornell, S., Bronzi, M., Zhong, J.: Attention is all you need in speech separation. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 21–25. IEEE (2021)
19. Sun, Y., Wang, X., Tang, X.: Deeply learned face representations are sparse, selective, and robust. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2892–2900 (2015)
20. Takeda, R., Komatani, K.: Unsupervised adaptation of deep neural networks for sound source localization using entropy minimization. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2217–2221. IEEE (2017)
21. Takeda, R., Kudo, Y., Takashima, K., Kitamura, Y., Komatani, K.: Unsupervised adaptation of neural networks for discriminative sound source localization with eliminative constraint. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) pp. 3514–3518 (4 2018). https://doi.org/10.1109/ICASSP.2018.8461723
22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
23. Wijmans, E., Kadian, A., Morcos, A., Lee, S., Essa, I., Parikh, D., Savva, M., Batra, D.: Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. arXiv preprint arXiv:1911.00357 (2019)
24. Zhang, Z., He, B., Zhang, Z.: Transmask: A compact and fast speech separation model based on transformer. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5764–5768. IEEE (2021)
25. Zhao, H., Gan, C., Rouditchenko, A., Vondrick, C., McDermott, J., Torralba, A.: The sound of pixels. In: Proceedings of the European conference on computer vision (ECCV). pp. 570–586 (2018)