Learning Audio-Visual Dynamics Using Scene Graphs for Audio Source Separation

—Supplementary Materials —

Moitreya Chatterjee[1*] **Narendra Ahuja**[1] **Anoop Cherian**[2*]
`metro.smiles@gmail.com` `n-ahuja@illinois.edu` `cherian@merl.com`
[1]University of Illinois, Urbana-Champaign, Urbana, IL
[2]Mitsubishi Electric Research Labs, Cambridge, MA

1 Summary of Supplementary Results

In this supplementary document, we provide:

- Details of the ASIW and AVE datasets
- Studies on generalizability of our approach
- Ablation studies including compute time analysis
- Ablation studies on the choice of the RBF kernel scale
- User study on the quality of the separated audio
- Details of our network architecture
- Details of our compute environment, and
- Qualitative results on direction prediction.

We also provide (in the supplementary bundle):

- **Video demonstration** of the audio separation capability of our approach. Please use a standard media player like VLC or Microsoft Movies & TV for playing it.

2 Details of the ASIW and AVE Datasets

Audio Separation in the Wild (ASIW) Dataset: The *Audio Separation in the Wild* (ASIW) dataset [1] consists of 147 validation, 322 test, and 10,540 training videos crawled from the larger AudioCaps dataset [5]. It has 14 auditory classes which are: *baby, bell, bird, camera, clock, dogs, toilet/drain, horse, man/woman, telephone, train, sheep/goat, vehicle/bus, water/water tank*. Each video in the dataset is 10s long, has significant camera motion, and captures diverse audio-visual contexts. For the direction prediction task, we divide each video into temporal windows of 1s long consisting of 8 frames each and predict the direction of motion for a randomly chosen window.

Audio Visual Event (AVE) Dataset: The popular *Audio Visual Event* (AVE) Dataset [11] was originally designed for the task of identifying audio-visual events. We adapt it to evaluate our approach. We treat each audio-visual event class as an auditory class and associate with it a set of *Auditory Objects*. The dataset contains 2211 training, 257 validation, and 261 test set videos. We use videos corresponding to 18 audio-visual event classes, including both potentially-moving and stationary object classes, for which pre-trained object detectors are avalable. The classes used for our experiments are: *man, woman, car, plane, truck, motorcycle, train, clock, baby, bus, horse, toilet,*

---

[*] Equal Contribution.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

violin, fiddle, flute, ukulele, acoustic guitar, banjo, accordion. The videos in this dataset are 10s long as well. Similar to ASIW, we divide each video into windows of 1s long (i.e. 8 frames per window) for computing the motion displacement vectors and predict the direction of motion for a randomly chosen window.

3 Generalizability of ASMP

To understand the real-world generalization of our approach to audio separation in videos outside the datasets used for training the model, we downloaded several videos from Youtube each with an arbitrary mix of multiple sounds, with the goal of applying our approach on them to separate the sounds. We selected videos having audio in the set of classes of the ASIW dataset. We then applied the model trained on the ASIW dataset on these videos. The results for these experiments are provided in the supplementary video. Our results clearly demonstrate that our approach leads to high quality audio separation even when the scene contains heavy distractors, such as continuous sound of rain, jammed traffic, or weak sounds, such as train announcements, etc. and exhibits reasonable amount of out of domain generalizability.

4 Additional Ablation Results

In this section, we provide additional studies on the importance of each component in our model. We note that some of these results are already in the main paper, however are grouped in a different manner.

4.1 3D Graph and Direction Prediction

In Table 1, we examine the contribution of each of the core novelties of our method on the ASIW dataset. **From the table, we see that the absence of the *direction prediction loss* supervision deteriorates the SDR performance by 0.6 dB**. Given that SDR is in log-scale, this implies a significant drop in audio separation quality, thereby attesting to the importance of this loss in the separation task. In order to assess the contribution of the *Pseudo 3D Scene Graph*, we replaced our 3D scene graph with a 2D scene graph (as in AVSGS [1]) , where the edges do not denote 3D spatial proximity, instead assume a fully-connected graph structure with equally-weighted edges. From the table, we see that this results in a drop of about 0.2 dB in the SDR, emphasizing the criticality of this module. However, when the direction prediction loss is introduced back into the training regime, we notice a slight boost in performance.

Table 1: SDR, SIR, SAR performance for different ablated model components of ASMP on the ASIW test set. [Best results in **bold**.]

		ASIW		
	Method	SDR↑	SIR↑	SAR↑
1.	ASMP (Full Model)	**9.6**	**14.5**	**14.1**
2.	ASMP: w/o Direction Prediction ($\lambda_4 = 0$)	9.0	14.3	13.7
3.	ASMP: w/o 3D Spatial Distances & w/o Direction Prediction	8.8	14.1	13.0
4.	ASMP: w/o 3D Spatial Distances & w/ Direction Prediction	9.1	14.3	13.6

4.2 Ablations on Losses

Table 2 presents the results of ablating the different loss terms in our optimization objective on the ASIW dataset. The results reveal that the direction prediction loss alone, while crucial, is insufficient in singularly providing a reasonably accurate separation/direction prediction performance. However, we find that training the models with any one of the other loss terms in conjunction with the direction prediction loss, results in a boost in model performance, with the presence of the co-separation loss resulting in the maximum gain. This underscores the importance of all three loss terms: co-separation

Table 2: SDR, SIR, SAR, and Direction Prediction performance for loss ablated models on the ASIW test set. [Best results in **bold**.]

		ASIW			
	Method	SDR↑	SIR↑	SAR↑	10-class Dir. Acc. (in %) ↑
1.	ASMP (Full Model)	**9.6**	**14.5**	**14.1**	**42.5**
2.	ASMP- Only Direction Prediction ($\lambda_1 = \lambda_2 = \lambda_3 = 0$)	6.4	11.2	11.7	32.7
3.	ASMP- Direction Prediction + $\mathcal{L}_{\text{ortho}}$ ($\lambda_1 = \lambda_2 = 0$)	6.4	11.7	10.1	33.2
4.	ASMP- Direction Prediction + $\mathcal{L}_{\text{co-sep}}$ ($\lambda_1 = \lambda_3 = 0$)	7.9	13.2	10.9	35.6
5.	ASMP- Direction Prediction + $\mathcal{L}_{\text{cons}}$ ($\lambda_2 = \lambda_3 = 0$)	6.4	11.6	11.9	33.1
5.	ASMP- No Direction Prediction ($\lambda_4 = 0$)	9.0	14.3	13.7	-

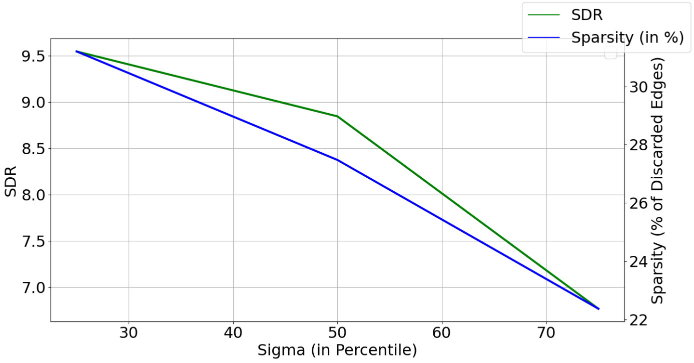

Figure 1: A plot showing the sensitivity of our proposed ASMP model to the choice kernel bandwidth (percentile) against the sparsity induced into the graph.

loss, consistency loss, and the orthogonality loss, besides the direction prediction loss for effective model training.

4.3 Compute Time Analysis

Thanks to the use of GPUs, the Chamfer distance between pairs of point clouds can be computed very quickly. Typically, a forward pass of one batch (with 25 samples) through the full network takes about 1.2 seconds on a Intel Core i7 workstation with NVIDIA RTX 2070 GPUs. When the additional task of computing the Chamfer distance is not there, i.e. the 3D Scene Graph is replaced with a graph with equal edge weights, this results in a saving of only 0.06 seconds per batch.

5 Performance Against Varying Kernel Scale

Recall that the Radial Basis Function (RBF) kernel is used to model the weighted adjacency matrix of the 3D scene graph, where the kernel is constructed on the Chamfer distances between 3D coordinates of the graph nodes. Thus, the bandwidth of the kernel describes a *soft ball* of a certain radius around each node to accommodate other nodes in the graph. As this radius depends on the specifics of the 3D graph constructed on the scene, it is not ideal to use a fixed radius for all graphs. Instead, we first compute a Chamfer distance matrix across all pairs of nodes in a given graph, and use the σ-th percentile of these distances to form the kernel bandwidth. In Figure 1, we plot the model performance (SDR) against increasing RBF kernel bandwidth, as well as the sparsity in the resulting graph (y-axis on the right), assessed by counting the number of edges with weights below $1e-5$ as a fraction of the total number of edges, which we discard. In the main paper, we reported results using the 25^{th}-percentile of the Chamfer distance matrix, which corresponds to $\sigma = 25\%$. In Figure 1, we plot against $\sigma = 25, 50, 75$ as well. Note that using a a higher σ leads to several dense connections, resulting in lower sparsity, leading to mistaken contexts for a node for separation. We found that using $\sigma = 25\%$ leads to the best performance.

Table 3: Human preference score on samples obtained from our method vs. AVSGS [1]

Datasets	**Prefer ours**
ASIW - Ours vs. AVSGS [1]	**63%**
AVE - Ours vs. AVSGS [1]	**68%**

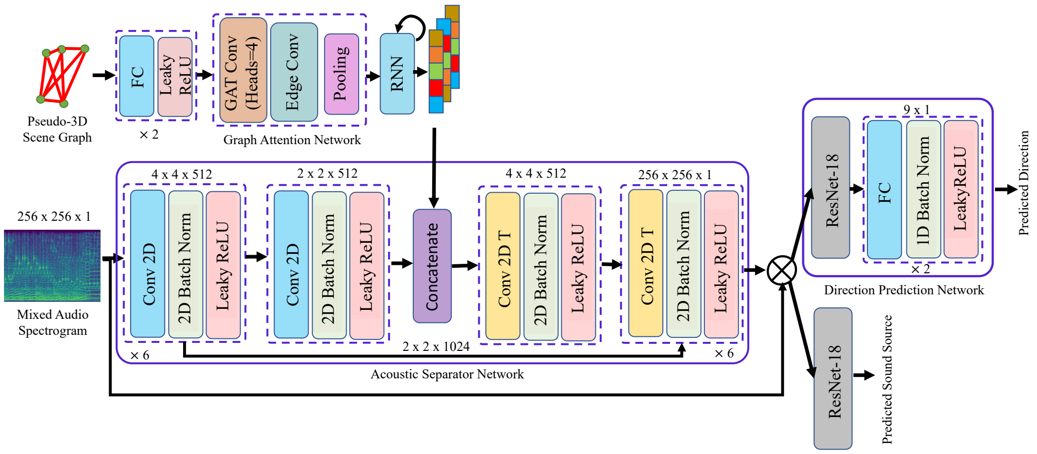

Figure 2: A detailed illustration of our proposed ASMP model.

6 User Assessment Study

In order to subjectively assess the quality of audio source separation, we evaluated a randomly chosen subset of separated audio samples from ASMP and our closest competitor AVSGS [1] for human preferability on both ASIW and AVE datasets. For the purpose of this study, we chose a set of in-house annotators, who have successfully completed annotation/evaluation tasks for speech/audio separation, in the past. Table 3 reports these performances, which show a clear preference of the evaluators, for our method over AVSGS about 60–65% of the time, on average.

7 Details of Compute Environment

We conduct experiments on a cluster of workstations, each with Intel Core i7 CPU, with 256GB RAM. Each of the workstations is equipped with 8 NVIDIA RTX 2070 GPUs.

8 Network Architecture Details

Our model, the *Audio Separator and Motion Predictor* (ASMP) has several components, as shown in Figure 2. Below, we list the key architectural details of each of them.

8.1 Feature Extractor

Our model, ASMP, captures the visual representation of the objects in the scene by means of *Pseudo-3D* scene graph; representing the visual entities of the scene as nodes of a graph (with associated features). These features are computed using a Faster R-CNN (FRCNN) network [9], with a ResNet-101 [3] backbone pre-trained on the Visual Genome Dataset [7]. The features of the musical instrument type nodes that appear in the Audio Visual Event (AVE) dataset are obtained by training another FRCNN network on the images of musical instruments on the OpenImages dataset [6]. The FRCNN network yields 2048-dimensional vectors for the Visual Genome dataset, which are then mapped to 512-dimensions via a 2-layer Multi-Layer Perceptron (MLP) with Leaky ReLU activations (negative slope=0.2), while the network yields 512-dimensional vectors for object features for the images in the OpenImages dataset and are used as is.

8.2 Pseudo-3D Graph Attention Network

Post the object detection and feature extraction, we construct the pseudo-3D scene-graph following the method laid out in the *Proposed Method* section of the paper. The scene graph is processed by a *Graph Attention Network*, which is a cascade of the following three modules:

Graph Attention Network Convolution (GAT Conv): This module [12] is responsible for updating the node features of the graph by employing a multi-headed graph message passing based on the adjacency of the nodes and edge weights. Our graphs are fully-connected and the edge weights are determined by the RBF kernel score as discussed in the paper. We design a 4-headed network, which outputs a 512-dimensional vector that embeds the full graph.

Edge Convolution (Edge Conv): Edge Convolutions [13] act on the output of the Graph Attention Network modules. These take in a concatenated pair of features associated with the two nodes of an edge and produces a vector as output. The input dimensions are thus $512 \times 2 = 1024$, while the output dimension is 512.

Pooling Layers: Finally, the *Graph Attention Network* pools the updated features [8] across the nodes obtained from the previous steps. The Global Max and Average Pooling techniques are used for this purpose and the feature vectors from the two are concatenated.

8.3 Recurrent Network

To iteratively (and automatically) segment the graph into appropriate subgraphs, we introduce a Recurrent Network into the pipeline. We instantiate it using a *Gated Recurrent Unit* (GRU) [2], whose input space and feature dimensions are 512-dimensional.

8.4 Acoustic Separator Network

The actual task of separating a mixed audio into its constituents is undertaken by an encoder-decoder network called the *Acoustic Separator Network*. In broad strokes, the network follows a U-Net [10] style architecture, with an encoder, bottleneck, and decoder modules. The encoder and decoder parts consists of 7 convolution and 7 up-convolution layers, respectively. Each layer has 4×4 filters with LeakyRELU activations and negative slope of 0.2. Moreover, encoder layers with matching spatial resolution of output feature maps are connected by skip connections to the corresponding decoder layers. The bottleneck layer concatenates the encoder embedding with the scene graph embedding, with each element of the latter tiled 2×2 times. The dimensions of the bottleneck layer are $(2 \times 2 \times 512)$, each for the encoder embedding and the scene graph embedding.

8.5 Direction Prediction Network

A central innovation in ASMP is the aspect of using the motion direction of a sound source as an additional supervisory signal to train the audio separation module. In order to leverage this auxiliary supervision, we take time-slices of the separated source spectrogram and pass it through a network called the *Direction Prediction Network*, which predicts which one of the 8 (or 26 depending on the setting) direction of motions (and one additional direction denoting no motion, and a second additional class denoting the displacement of the background) centered at the current location of the object, did the object move in, within the window. The network consists of a ResNet-18 style module [3] for embedding the separated spectrogram into a 512-dimensional vector. This is followed by a a 2-layer Multi-Layer Perceptron (MLP). The hidden layer embedding is passed through a 1d BatchNorm [4] and Leaky ReLU activations (negative slope=0.2) before being processed by the second layer of the MLP. The output of the second layer is a 10 (or 28)-dimensional vector, constituting the logits of the direction prediction classifier.

9 Qualitative Results

In Figures 3, 5, and 4, we present several qualitative results demonstrating the direction prediction by ASMP across ASIW and AVE datasets. We show these results on the starting frame of a window used for the direction prediction, and show a sequence of such windows. Also shown is a cube centered at an *Auditory Object* in the scene, with one of its corners marked with a green dot denoting the

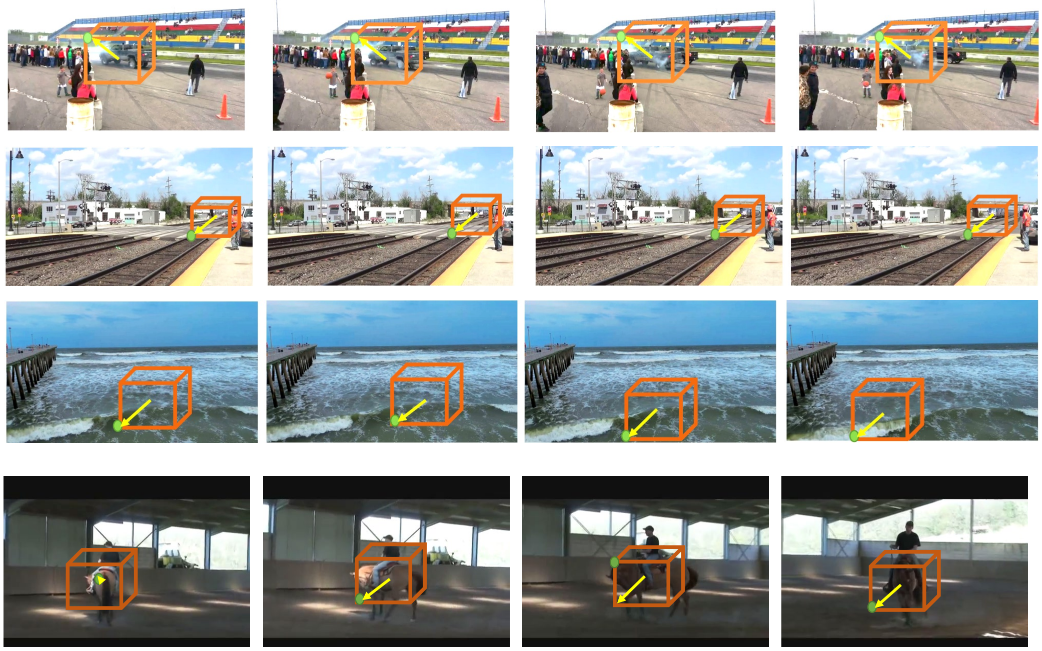

Figure 3: Qualititive results from the ASIW dataset demonstrating object 3D motion prediction using our method. The green dot shows the ground truth direction on the orange unit cube, and the yellow arrow shows the predicted direction. The unit cube is placed at the center of the object detection bounding box.

true direction of motion of the object in the window (as estimated using our optical flow estimation, between the first and the last frames in a window, and combined with the pseudo 3D depth estimation. Note that, if an object has no motion, the green dot is then placed at the center of the cube. The estimated motion directions are denoted by yellow-colored vectors and point to one of the eight corners of the cube or to the cube center (in case there is no motion).

References

[1] Moitreya Chatterjee, Jonathan Le Roux, Narendra Ahuja, and Anoop Cherian. Visual scene graphs for audio source separation. In *Proc. ICCV*, 2021.

[2] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, pages 770–778, 2016.

[4] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[5] Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. Audiocaps: Generating captions for audios in the wild. In *Proc. NAACL HLT*, pages 119–132, 2019.

[6] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, et al. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github. com/openimages*, 2(3):18, 2017.

[7] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual Genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vis.*, 123(1):32–73, 2017.

[8] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. Self-attention graph pooling. In *Proc. ICML*, pages 3734–3743, June 2019.

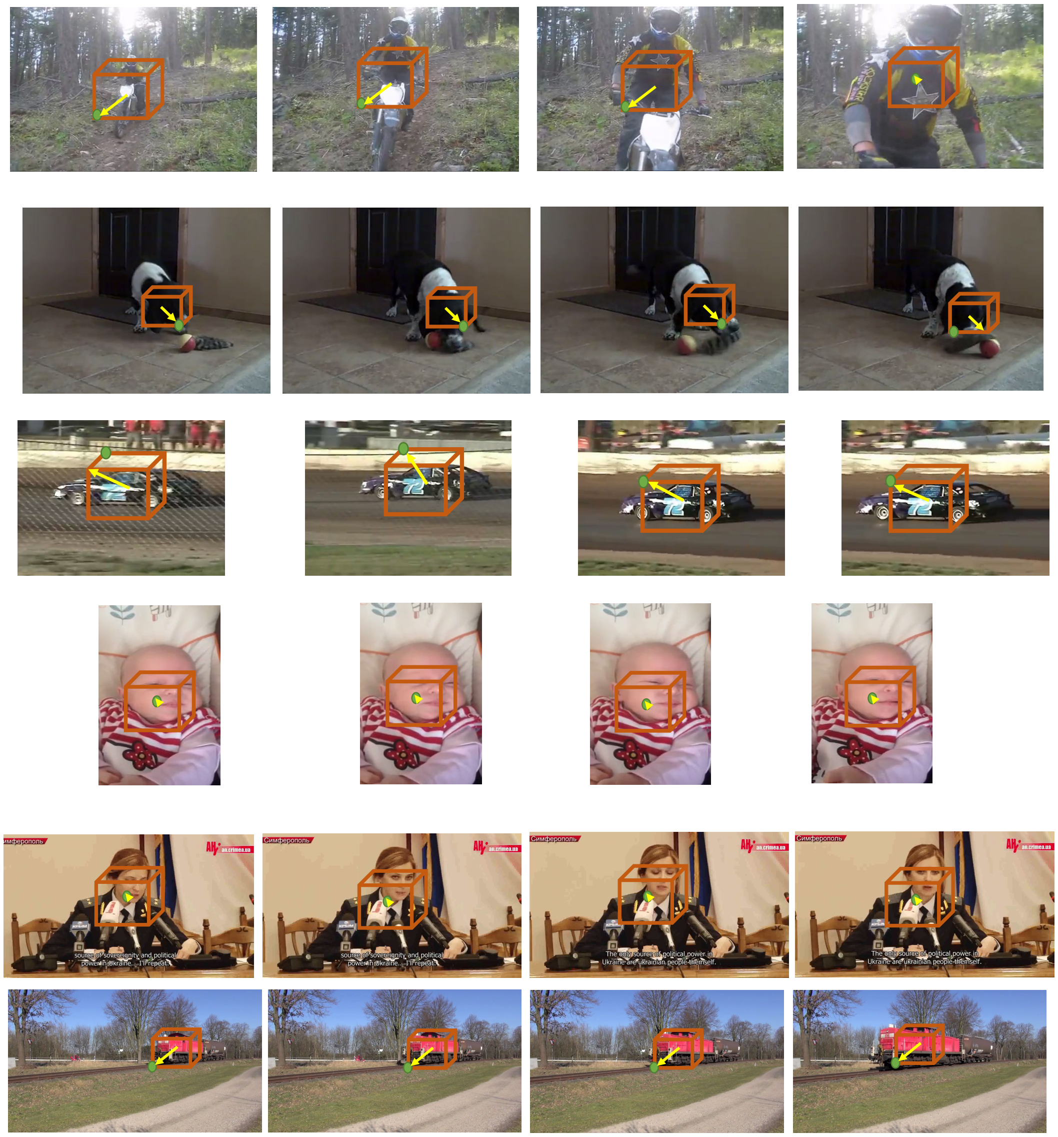

Figure 4: Additional qualitative direction prediction results on ASIW videos. We show a unit cube around the *Auditory Object* with a green dot denoting the ground truth direction of motion. The yellow arrow indicates the predicted motion direction by ASMP using only the audio signal from a mixed spectrogram.

[9] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2016.

[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proc. MICCAI*, pages 234–241. Springer, 2015.

[11] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *Proc. ECCV*, pages 247–263, 2018.

[12] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *Proc. ICLR*, April 2018.

[13] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph. (TOG)*, 38(5):1–12, 2019.

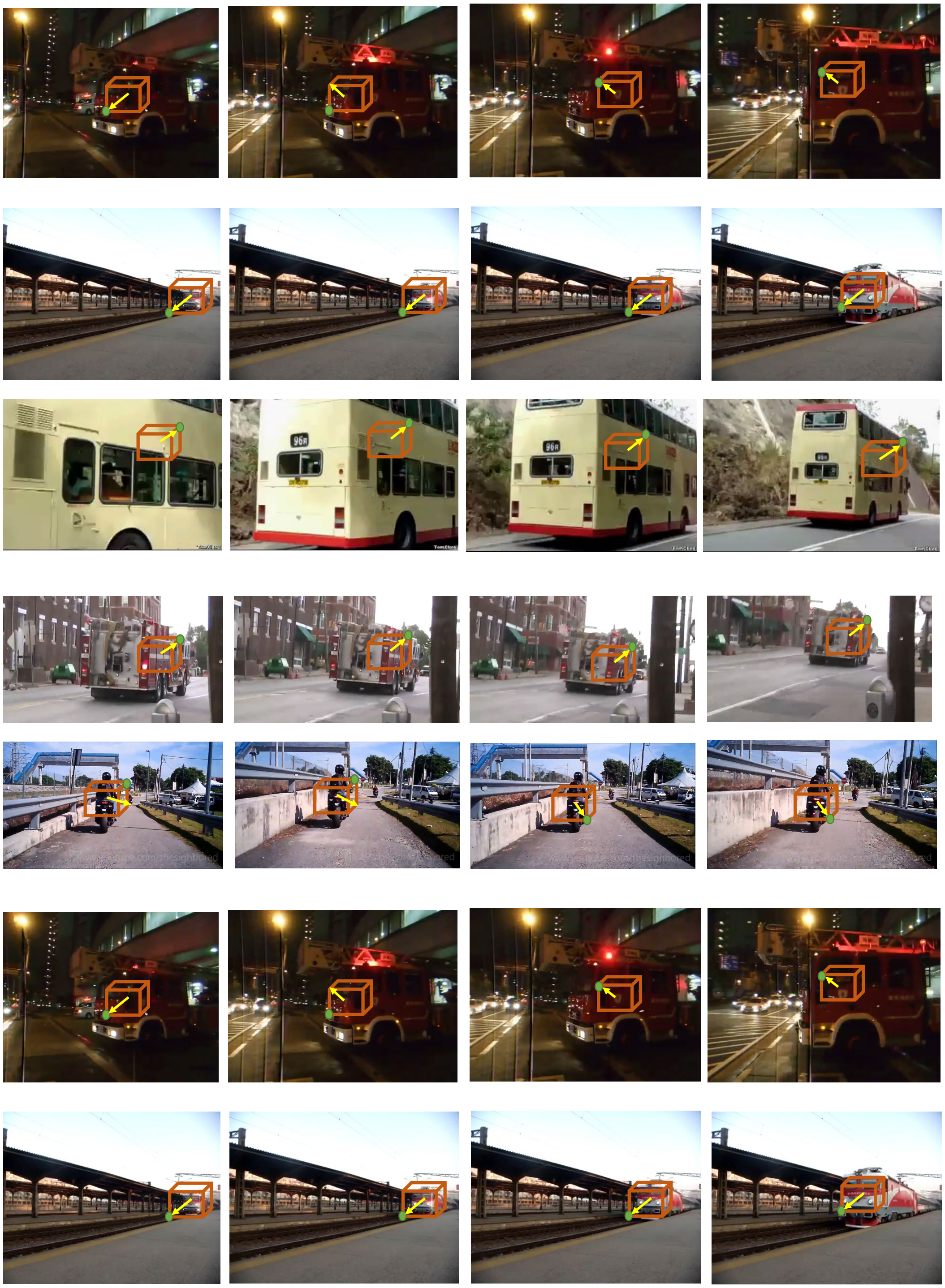

Figure 5: Qualitative direction prediction results on AVE videos. We show a unit cube around the *Auditory Object* with a green dot denoting the ground truth direction of motion. The yellow arrow indicates the predicted motion direction by ASMP using only the audio signal from a mixed spectrogram.