Supplementary Material: Do We Need Sound for Sound Source Localization?

Takashi Oya*, Shohei Iwase*, Ryota Natsume, Takahiro Itazuri, Shugo Yamaguchi, and Shigeo Morishima

Waseda Research Institute for Science and Engineering
oya_takashi@ruri.waseda.jp, sh.iwase@fuji.waseda.jp,
nano.poteto@toki.waseda.jp, s132800732@fuji.waseda.jp,
wasedayshugo@suou.waseda.jp, shigeo@waseda.jp

1 Unsupervised learning

Training procedure For the setting without pretrained weights for the image network, we use a fixed learning rate = 0.0001, with batchsize = 8 and epochs = 50. When we use pretrained weights for the image network, we use batchsize = 8, epochs = 25 and learning rate = 0.00001 for the image network, and learning rate = 0.0001 for the others. For all experiments, we use an Adam [1] optimizer.

Architecture details For the image network, we use VGG-11, implemented in pytorch [2]. We show the architecture of the image network in Table 1. For the sound network, we use a VGG-like architecture. Table 2 shows the architecture of the sound network.

For more details of our unsupervised model, refer to our code included in the supplementary materials.

2 Supervised learning

For the supervised training, we train an U-Net [3] with a fixed learning rate = 0.0001, batchsize = 8, epochs = 100, and the Adam optimizer. The encoder of the U-Net is ResNet-34 [4]. For More details of the architecture, refer to our code.

3 Saliency map

We use the output of Grad-CAM [6], which can be regarded as a class-specific saliency map. To obtain the saliency map for top-N classes, we use the max value for each pixel along N saliency maps, as follows.

$$S^N_{i,j} = \max_k S_{k,i,j}, \tag{1}$$

Table 1. The architecture of the image network. "conv" indicates a 2-dimensional convolutional layer, "size" is the size of the filters, and "n_filters" is the number of the filters.

Layer	Layer information
conv1	(n_filters = 64, size = 3, stride = 1, padding = 1), ReLU
maxpool1	(size = 2, stride = 2, padding = 0)
conv2	(n_filters = 128, size = 3, stride = 1, padding = 1), ReLU
maxpool2	(size = 2, stride = 2, padding = 0)
conv3	(n_filters = 256, size = 3, stride = 1, padding = 1), ReLU
conv4	(n_filters = 256, size = 3, stride = 1, padding = 1), ReLU
maxpool3	(size = 2, stride = 2, padding = 0)
conv5	(n_filters = 512, size = 3, stride = 1, padding = 1), ReLU
conv6	(n_filters = 512, size = 3, stride = 1, padding = 1), ReLU
maxpool4	(size = 2, stride = 2, padding = 0)

Table 2. The architecture of the sound network. "BatchNorm" means Batch Normalization [5].

Layer	Layer information
conv1	(n_filters = 32, size = 3, stride = 1, padding = 1), BatchNorm, ReLU
conv2	(n_filters = 32, size = 3, stride = 1, padding = 1), BatchNorm, ReLU
maxpool1	(size = 2, stride = 2, padding = 0)
conv3	(n_filters = 64, size = 3, stride = 1, padding = 1), BatchNorm, ReLU
conv4	(n_filters = 64, size = 3, stride = 1, padding = 1), BatchNorm, ReLU
maxpool2	(size = 2, stride = 2, padding = 0)
conv5	(n_filters = 128, size = 3, stride = 1, padding = 1), BatchNorm, ReLU
conv6	(n_filters = 128, size = 3, stride = 1, padding = 1), BatchNorm, ReLU
maxpool3	(size = 2, stride = 2, padding = 0)
conv7	(n_filters = 128, size = 3, stride = 1, padding = 1), BatchNorm, ReLU
conv8	(n_filters = 128, size = 3, stride = 1, padding = 1), BatchNorm, ReLU
maxpool4	(size = 2, stride = 2, padding = 0)
conv9	(n_filters = 128, size = 3, stride = 1, padding = 1), BatchNorm, ReLU
conv10	(n_filters = 128, size = 3, stride = 1, padding = 1), BatchNorm, ReLU

where $S^N_{i,j}$ is the saliency map for top-N classes and $S_{k,i,j}$ is the saliency map for the k-th predicted class.

We test the performances of the saliency maps for top-N classes, for N = 1, 10, 20, 50, 100, 200, 500, 1000 and find N = 100 works best; thus we report the result of N = 100 in the original paper. Table 3 shows the performances of saliency maps using different N.

4 Additional comparison with other methods

Though our main focus is to analyze the contribution of image/sound modalities, we made further comparisons with other various methods [8–10]. The results in 4 show that our method achieves the best performance. Scores for localization

Table 3. Evaluation of saliency maps for top N classes for varying N. cIoU score with threshold 0.5 and AUC score are reported.

N	saliency map for top-N classes	
	cIoU	AUC
1	45.7	41.1
10	51.5	45.1
20	52.1	45.7
50	52.6	46.2
100	52.7	46.3
200	52.5	46.1
500	52.4	46.0
1000	51.8	45.5

Table 4. Additional comparison with other methods. Scores for localization maps obtained using models trained in an unsupervised setting with 10k samples are reported.

Method	cIoU	AUC
Ours	56.8	50.7
Senocak et al. [7] (reported)	43.6	44.9
DMC [8] (reported)	41.6	45.2
CAVL [9] (reported)	50.0	49.2
Two-stage [10] (reported)	52.2	49.6

maps obtained using models trained in an unsupervised setting with 10k samples are reported. As the original paper of DMC [8] only report scores for models trained with 400k samples, we use the scores of DMC trained on 10k samples reported in CAVL [9].

5 Ablation Study

We conducted ablation study to further analyze our proposed model. Specifically, we trained our model without the potential localization network and the selection module, and compared their performances. The altered model is obtained by fixing the potential localization map to be a constant ($P_{i,j} = const.$), which is inherently the same as Senocak et al. [7]. The quantitative results are shown in Table 5. As a result, the models without the potential localization network and the selection module worked as good as Senocak et al. [7], but worse than our proposed method, which means the potential localization network and the selection module gives a positive effect on the performance of the localization

Table 5. Evaluation of localization maps. cIoU score with threshold 0.5 and AUC score are reported. "PLN, SM" denotes the potential localization network and the selection module. All the experiments are conducted in an unsupervised setting.

# of training samples	[7] (reported)		pretrained		pretrained & without PLN, SM		de novo		de novo & without PLN, SM	
	cIoU	AUC	cIoU	AUC	cIoU	AUC	cIoU	AUC	cIoU	AUC
1k	—	—	48.7	46.4	44.4	40.6	36.5	34.1	35.7	33.0
2.5k	—	—	50.3	47.7	48.7	45.8	40.7	37.3	38.6	36.6
10k	43.6	44.9	56.8	50.7	56.4	49.2	48.4	45.3	47.6	44.3
144k	66.0	55.8	68.4	57.0	66.8	55.7	66.7	56.3	65.8	55.3

Random	cIoU	AUC
	34.1	32.3

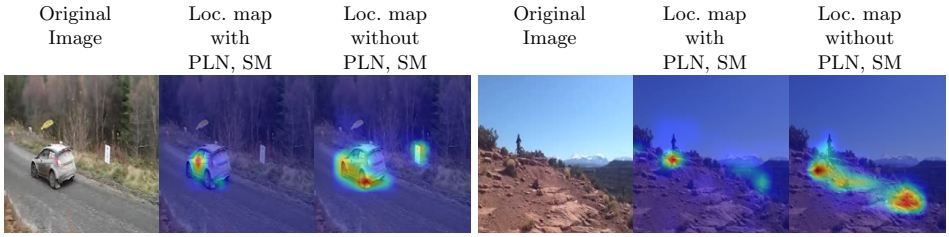

Fig. 1. Visualization of localization maps for the models with/without the potential localization network (PLN) and the selection module (SM). The model without PLN and SM, which is inherently same as Senocak et al. [7], responds to objects that can not produce sound (e.g. signboard), while the model with PLN and SM does not. For the visualization, we used the models trained with 144k samples and pretrained weights.

map. This result suggests that the potential localization network allows efficient training as the potential localization map $P_{i,j}$ predetermines the possible sound sources (i.e. only responds to objects that can produce sound).

Why does $P_{i,j}$ only respond to objects that can produce sound?
Though our models does not have any constraints that prevents the potential localization map from being a constant, the potential localization map $P_{i,j}$ only responds to objects that can produce sound. We believe this is because of the way the localization map is obtained (Eq. (3)). If $P_{i,j} = const.$, the model can only use $A_{i,j}$ to locate the sound source. However, if $P_{i,j}$ functions like a prior distribution and predetermines the possible sound sources (i.e. eliminates the objects that can not produce sound), the workload of $A_{i,j}$ is reduced, allowing the model to be trained more easily. This mechanism motivates $P_{i,j}$ to not be a constant, but rather the possible sound sources. As seen in Fig. 1, the altered model responds to objects that can not produce sound, while our model does not.

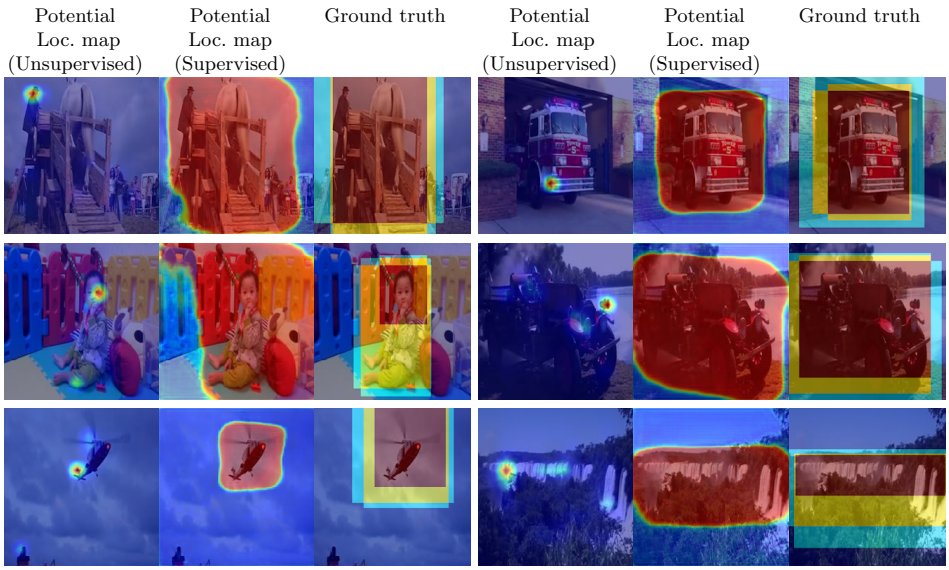

Fig. 2. Visualization of potential localization maps for our unsupervised model and supervised model. For the unsupervised setting, we used the model trained with 144k samples and pretrained weights.

6 Comparison between our unsupervised model and supervised model

Visualization of potential localization maps for our unsupervised model and supervised model is shown in Fig. 2. It can be noted that for some samples, the potential localization map of the unsupervised model only responds to a small part of the observed object, whereas the supervised model mostly encompasess the entire object. This can be attributed to the fact that, as with existing unsupervised methods, our unsupervised model does not have any contraints that force the localization map and potential map to exhibit this trait. The model is only trained to minimize the similarity loss which does not necessarily require the model to respond to the entire object.

7 Annotation Details

The annotation process of the dataset used to analyze the performance gap in Table 4 in the original paper is as follows. First, we randomly obtained 4K samples from Flickr-SoundNet [11] excluding the samples contained in the benchmark dataset [7]. Then, we searched Type-B samples (there are several objects capable of producing sound in the image, but of which only one object is actually producing sound), by manually checking each image-sound pair. It should be noted that we had to listen to the sound in the annotation process to decide

whether the samples are Type-A or Type-B, as their distinction is dependent on the accompanying sound as stated above. As a result, we found 30 Type-B samples in this process. We obtained the same number of Type-A samples to match the number of each sample. Finally, we annotated sound source for each sample using bounding boxes. In Fig. 3 and Fig. 4, we show the examples of Type-A and Type-B images, and sound source annotations for them.

8 Video

The supplementary materials contain **"video.mp4"**. This video shows 3 cases where the localization map and the potential localization map are different. For instrument and machine sound, we use a 5-second clip from the original audio. For human sound, we use a 5-second clip from another video's sound that only contains human sound. For the visualization, we use the model trained with 144k samples and pretrained weights.

9 Code

The supplementary materials contain a jupyter notebook file **"code/code.ipynb"** and a html file **"code/code.html"**. The contents of these files are the same, but **"code/code.html"** does not require an environment for jupyter notebook. These files contain main parts of our code, including the network architectures and how we obtain the saliency maps. We also include **"code/requirements.txt"**, which shows the list of libraries we use.

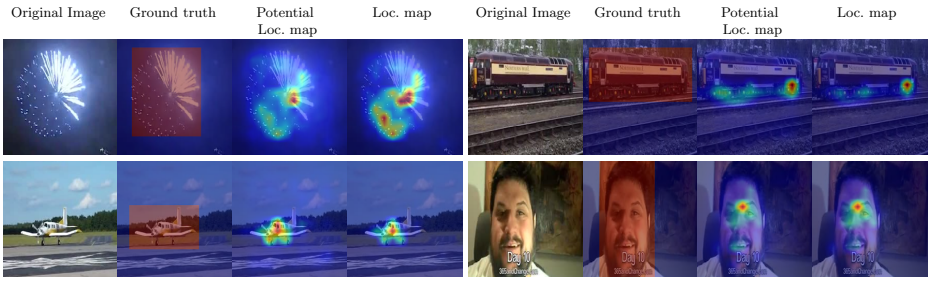

Fig. 3. Visualization of potential localization maps and localization maps of Type A along with the ground truth. For the visualization, we used the model trained with 144k samples and pretrained weights.

Fig. 4. Visualization of potential localization maps and localization maps of Type B along with the ground truth. For the visualization, we used the model trained with 144k samples and pretrained weights.

References

1. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR). (2015) 1
2. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. In: Neural Information Processing Systems Workshops (NeurIPSW). (2017) 1
3. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI). (2015) 1
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Computer Vision and Pattern Recognition (CVPR). (2016) 1
5. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning (ICML). (2015) 2
6. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: International Conference on Computer Vision (ICCV). (2017) 1
7. Senocak, A., Oh, T.H., Kim, J., Yang, M.H., So Kweon, I.: Learning to localize sound source in visual scenes. In: Computer Vision and Pattern Recognition (CVPR). (2018) 3, 4, 5

8. Hu, D., Nie, F., Li, X.: Deep multimodal clustering for unsupervised audiovisual learning. In: Computer Vision and Pattern Recognition (CVPR). (2019) 2, 3
9. Hu, D., Wang, Z., Xiong, H., Wang, D., Nie, F., Dou, D.: Curriculum audiovisual learning. arXiv preprint arXiv:2001.09414 (2020) 2, 3
10. Qian, R., Hu, D., Dinkel, H., Wu, M., Xu, N., Lin, W.: A two-stage framework for multiple sound-source localization. In: Computer Vision and Pattern Recognition Workshops (CVPRW). (2020) 2, 3
11. Aytar, Y., Vondrick, C., Torralba, A.: Soundnet: Learning sound representations from unlabeled video. In: Neural Information Processing Systems (NeurIPS). (2016) 5