– Supplementary Material –

Event-Specific Audio-Visual Fusion Layers: A Simple and New Perspective on Video Understanding

Supplementary Material

In this supplementary material, we present additional details and results pertaining to the experiments that are not included in the main text due to space constraints. All figures and references in this supplementary material are self-contained. The contents included in this supplementary material are as follows: 1) Details of the backbone network architectures, audio preprocessing, and datasets, 2) Layer visualizations, 3) Layer analysis with VGGSound categories, 4) Additional sound localization results, 5) Potential applications, 6) Annotation process for the VGGSound multilabel analysis.

A. Architecture Details of Backbone Networks

In Table 1, we provide the architecture of the backbone networks. We use two-stream network architecture, video network and audio network, as in existing audio-visual learning works. The video network is a spatio-temporal ResNet mixed convolution network, identical to MCx [11], borrowed from official PyTorch implementation[1] (mc3_18). The audio network architecture is similar to [1]. Batch Normalization and ReLU activation function are used after every convolution layer.

The networks are trained using the SGD optimizer with the starting learning rate of 0.01, and the learning rate is reduced by a factor of 10 if the validation accuracy does not increase for 3 epochs.

B. Audio Preprocessing Details

Our audio preprocessing follows the setting used in the previous works [1, 12]. We sample audio data with 16kHz sampling rate and input audio is 10 seconds. STFT is computed using $n\_fft = 512$, $hop\_length = 160$, $win\_length = 320$, $window = hann\_window(320)$, $center = True$ and $pad\_mode = reflect$ and 1000×80 log-mel spectrogram is produced with 80 mel filterbanks by using PyTorch. Audio onsets are computed using librosa [6] onset detection function with a pre-computed onset envelopes.

C. Datasets

We train and validate our method on five video datasets using standard evaluation metrics. **VGGSound** [4] is a recently released audio-visual dataset which contains around ∼200K videos obtained from YouTube and labelled with 309 categories. **Kinetics-400** [5] is a large-scale standard benchmark dataset for action recognition with ∼240K training and 20K validation videos containing 400 human action classes. **Kinetics-Sound** [2] is created by choosing 34 classes from Kinetics dataset that are assumed to have audio and visual characteristics and it has total ∼22k videos. **AVE** [10] is another audio-visual dataset formed for audio-visual event localization and it contains around 4K videos covering 28 event categories. **LLP** [9] is a multi-label dataset consisting of ∼12k videos labeled by 25 categories and formed for audio-visual video parsing.

However, some of the videos are removed or not accessible from the web because of privacy or regional settings. Hence, our datasets may be slightly smaller than official numbers for some datasets (See Table 2). Additionally, the original 34 classes in Kinetics-Sound are based on the earlier version of the Kinetics. Some classes are removed currently. Therefore, we use available 31 classes.

For multi-label evaluation, we construct an additional subset of VGGSound, called Multi-labeled VGGSound. Please refer to Section G.

[1] https://pytorch.org/vision/0.8/models.html#torchvision.models.video.mc3_18

Table 1: **Architecture of the backbone networks.** K, S, P, res, $maxpool$ and $avgpool$ denote kernel size, stride, padding, residual, max-pooling and average-pooling layers, respectively.

Layer	# filters	K	S	P	Output
input	1	-	-	-	$10 \times 100 \times 80$
conv1	64	(1,3,3)	(1,2,1)	(0,1,1)	$10 \times 50 \times 80$
conv2	64	(1,3,3)	(1,1,2)	(0,1,1)	$10 \times 50 \times 40$
maxpool2	-	(1,1,3)	(1,1,2)	(0,0,0)	$10 \times 50 \times 19$
conv3	192	(1,3,3)	(1,1,1)	(0,1,1)	$10 \times 50 \times 19$
maxpool3	-	(1,3,3)	(1,2,2)	(0,0,0)	$10 \times 24 \times 9$
conv4	256	(1,3,3)	(1,1,1)	(0,1,1)	$10 \times 24 \times 9$
conv5	256	(1,3,3)	(1,1,1)	(0,1,1)	$10 \times 24 \times 9$
conv6	256	(1,3,3)	(1,1,1)	(0,1,1)	$10 \times 24 \times 9$
maxpool6	-	(1,3,2)	(1,2,2)	(0,0,0)	$10 \times 11 \times 4$
conv7	512	(1,4,4)	(1,1,1)	(0,1,0)	$10 \times 10 \times 1$
fc8	512	(1,1,1)	(1,1,1)	(0,0,0)	100×1
fc9	512	(1,1,1)	(1,1,1)	(0,0,0)	100×1

(a) Audio Network

Layer	# filters	K	S	P	Output
input	3	-	-	-	$100 \times 112 \times 112$
conv1	64	(3,7,7)	(1,2,2)	(1,3,3)	$100 \times 56 \times 56$
conv2	64	(3,3,3)	(1,1,1)	(1,1,1)	$100 \times 56 \times 56$
conv3	64	(3,3,3)	(1,1,1)	(1,1,1)	$100 \times 56 \times 56$
conv4	64	(3,3,3)	(1,1,1)	(1,1,1)	$100 \times 56 \times 56$
conv5	64	(3,3,3)	(1,1,1)	(1,1,1)	$100 \times 56 \times 56$
conv6	128	(1,3,3)	(1,2,2)	(0,1,1)	$100 \times 28 \times 28$
conv7	128	(1,3,3)	(1,1,1)	(0,1,1)	$100 \times 28 \times 28$
res-conv8	128	(1,1,1)	(1,2,2)	(0,0,0)	$100 \times 28 \times 28$
conv9	128	(1,3,3)	(1,1,1)	(0,1,1)	$100 \times 28 \times 28$
conv10	128	(1,3,3)	(1,1,1)	(0,1,1)	$100 \times 28 \times 28$
conv11	256	(1,3,3)	(1,2,2)	(0,1,1)	$100 \times 14 \times 14$
conv12	256	(1,3,3)	(1,1,1)	(0,1,1)	$100 \times 14 \times 14$
res-conv13	256	(1,1,1)	(1,2,2)	(0,0,0)	$100 \times 14 \times 14$
conv14	256	(1,3,3)	(1,1,1)	(0,1,1)	$100 \times 14 \times 14$
conv15	256	(1,3,3)	(1,1,1)	(0,1,1)	$100 \times 14 \times 14$
conv16	512	(1,3,3)	(1,2,2)	(0,1,1)	$100 \times 7 \times 7$
conv17	512	(1,3,3)	(1,1,1)	(0,1,1)	$100 \times 7 \times 7$
res-conv18	512	(1,1,1)	(1,2,2)	(0,0,0)	$100 \times 7 \times 7$
conv19	512	(1,3,3)	(1,1,1)	(0,1,1)	$100 \times 7 \times 7$
conv20	512	(1,3,3)	(1,1,1)	(0,1,1)	$100 \times 7 \times 7$
avgpool	-	(1,7,7)	-	(0,0,0)	$100 \times 1 \times 1$

(b) Video Network

Dataset	Train	Test	Val.	Total
VGGSound	170384	0	13675	184059
Kinetics	208552	33595	17019	259166
Kinetics-Sound	19931	2677	1351	23959
AVE	3697	402	0	4099
LLP	9620	1162	624	11406

Table 2: **Dataset statistics in our experiments.**

D. Category-wise Layer Analysis

In Figure 1, we list the VGGSound categories that are assigned to each Audio-Visual event-specific layer of our network. To obtain these results, we apply majority voting rule among all the videos within each category and assign the layer label to the categories as we explain in "event-characteristics of a dataset" subsection of the main paper. We show some of these

categories (due to the limited space) for each audio-visual event layers in Figure 1.

The onset event layer predicts categories that are related to music, animal and repetitive actions or sounds as shown in Figure 1. As aforementioned in the main paper, musical instruments such as "playing tambourine", "playing steelpan", "beatboxing" or animal vocalization sounds – "frog croaking", "francolin calling", "chipmunk chirping" – and some actions such as "hammering nails", "forging swords", "smoke detector beeping" all tend to have rhythmic and repetitive characteristics. This aligns with our design motivation that onset event layer learns rhythmic, repetitive and periodic events as listed categories contain these characteristics.

The instant event layer focuses on the categories that contain impact events like "bowling impact", "closing car doors" or sudden events like "people nose blowing", "train horning" or explosion-kind of events such as "firing muskets", "splashing water" and "people farting". This also matches with our intuition that instant event layer predicts sudden, sparse highly audio-visual correlated instant events.

Finally, the categories that are highlighted by the continuous event layer have temporally constant-like characteristics such as "bathroom ventilation fan", "blowtorch igniting", "hair dryer drying" or slowly evolving sounds like "airplane" or "sea waves". The results also show that our intuition on the continuous event layer matches with these categories.

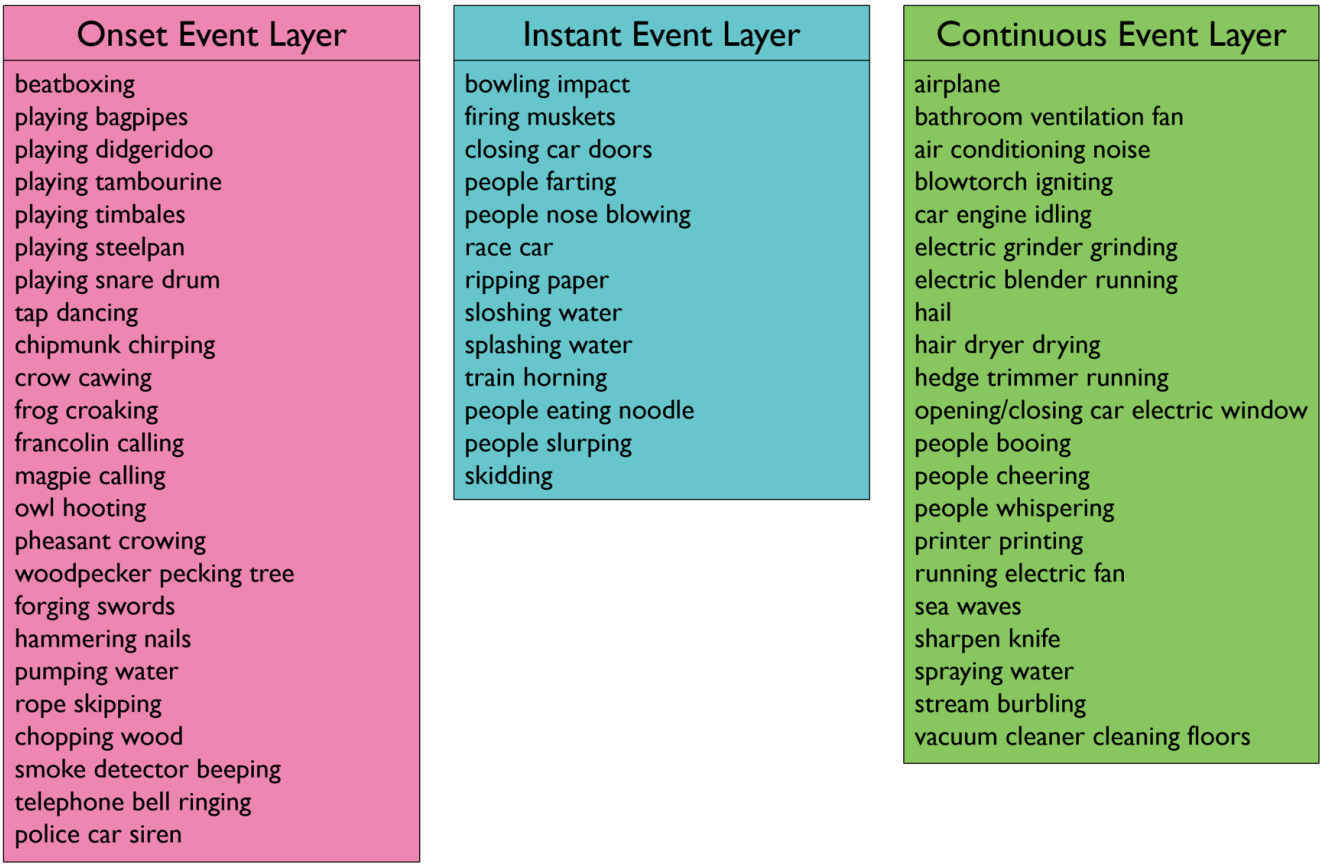

Figure 1: **VGGSound categories that are assigned to each audio-visual event layer.**

E. Sound Localization

Figure 2 shows the additional qualitative results of the sound localization attempt, spatially and time-wise, by using the features from our backbone networks throughout videos. We also visualize the attention maps of VGG-SS test samples in Figure 3 and compare them with the state- of-the-art [3] method. Figure 3 shows how two different approaches response to the same samples.

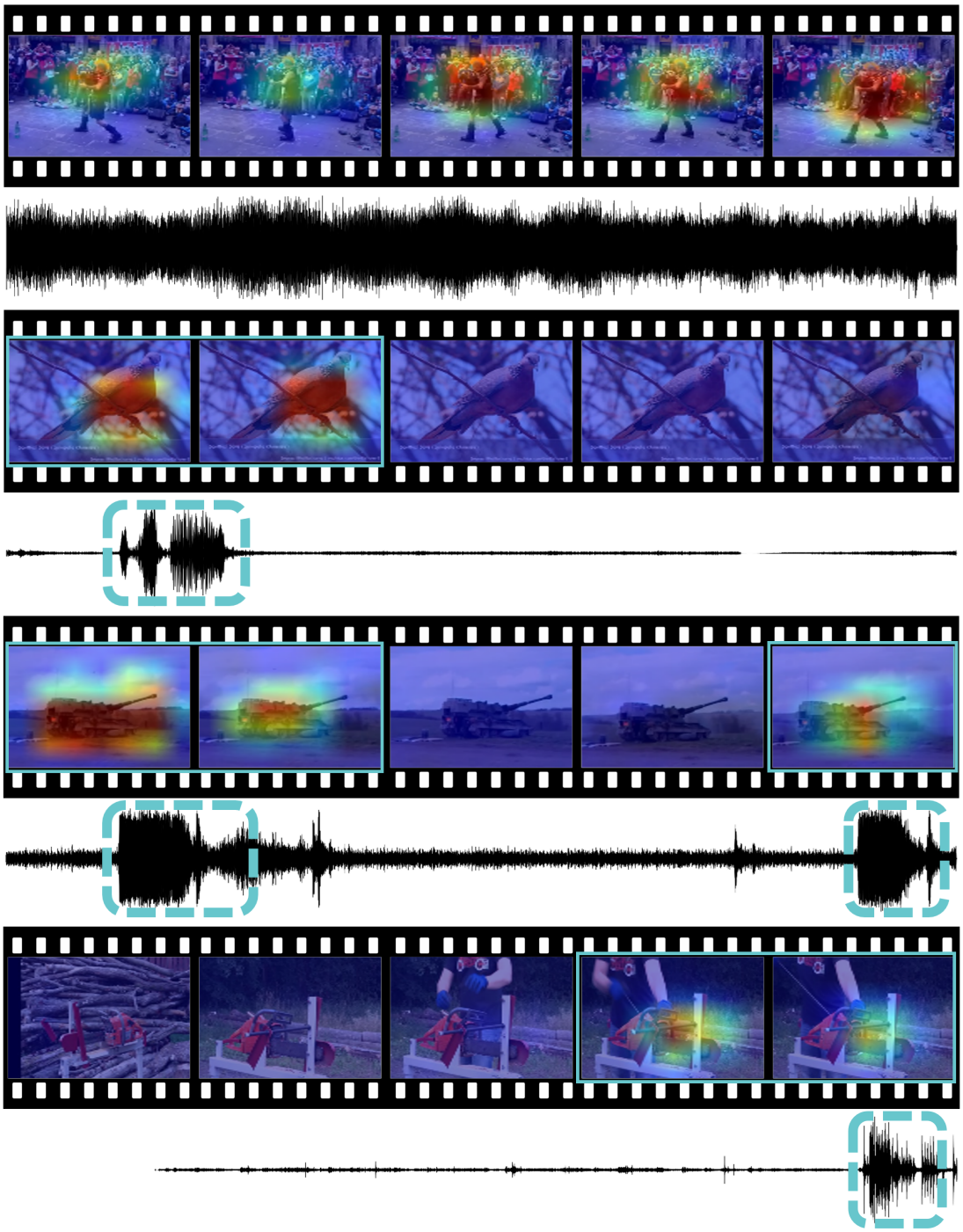

Figure 2: **Additional Sound Localization Results.**

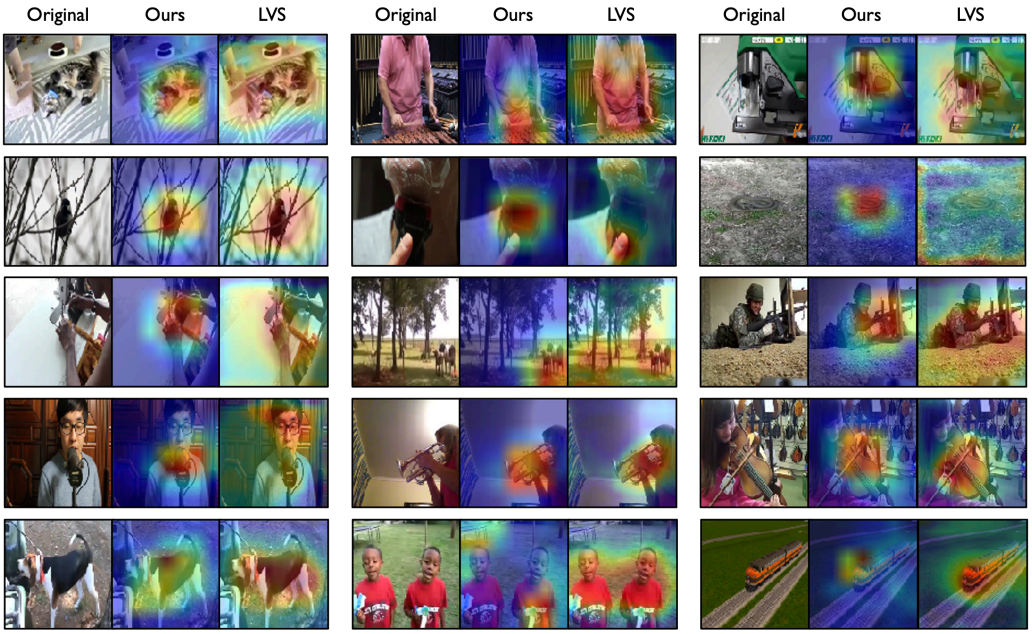

Figure 3: **Sound Localization Results on VGG-SS and comparison with LVS [3].**

Method	IoU	AUC	Main Task
AVEL [10]$_{\text{ECCV18}}$	0.291	0.348	×
AVobject [1]$_{\text{ECCV20}}$	0.297	0.357	×
Ours	0.309	0.354	×
Attention [7]$_{\text{CVPR18}}$	0.185	0.302	✓
LVS [3]†$_{\text{CVPR21}}$	0.303	0.364	✓
SSPL(w/o PCM) [8]$_{\text{CVPR22}}$	0.270	0.348	✓
SSPL(w/ PCM) [8]$_{\text{CVPR22}}$	0.339	0.380	✓

Table 3: **Sound source localization on the VGG-SS test set [3]**. All models are trained on the VGG-Sound 144k and tested on VGG-SS. † is obtained from the model provided by the original authors.

In Table 3, we quantitatively evaluate sound source localization capability. We use the recently released VGG-SS dataset [3] that is curated from VGGSound containing around 5k samples. Among the audio-visual learning models, our method performs favorably against the other audio-visual models [10, 1]. Moreover, our method shows competitive performance against the audio-visual models that are explicitly trained to tackle sound localization as a main task.

F. Potential Applications

In "Concluding Remarks" section of the main paper, we have discussed potential applications that can be built based on our model. In this supplementary material, we present some examples for these applications.

F.1. Dataset Retargeting / Cleanup

As an example to dataset retargeting application, we use our proposed method to identify the event characteristic distribution of Kinetics dataset and only select the categories that fall under the audio-visual event layers. Kinetics-Sound dataset is constructed to have a high efficacy in audio-visual learning tasks. Based on the subset we construct, we compare to see how many categories of Kinetics-Sound matches with the categories that our audio-visual event layers filter. This experiment reveals that 66% of the Kinetics-Sound categories are matched. Kinetics-Sound categories that intersect with the categories that our audio-visual event layers filtered from Kinetics are listed in Figure 4. The categories marked in red are the categories

of Kinetics-Sound that are not selected by our audio-visual event layers filter.

Kinetics-Sound Categories ∩ Our AV Categories in Kinetics			
blowing nose	bowling	chopping wood	ripping paper
singing	tapping pen	blowing out candles	dribbling basketball
moving lawn	shoveling snow	stomping grapes	tap dancing
tickling	playing accordion	playing bagpipes	playing bass guitar
playing drums	playing guitar	playing harmonica	playing keyboard
playing piano	playing saxophone	playing trombone	playing trumpet
playing xylophone	strumming guitar	tapping guitar	playing clarinet
shuffling cards	laughing	playing organ	playing violin

Figure 4: **Kinetics-Sound categories that intersect with our AV categories in Kinetics.**

F.2. Modality-level Video Understanding

We present an example of modality-level video understanding in Figure 5. As shown in the figure, the modality confidence of each layer is highly activated only when there is meaningful signal. The audio modality confidence level aligns with the audio signal presence. Similarly, vision modality confidence is activated when the meaningful frames, *i.e.*, frames with owl, appear. The audio-visual modality, which is per second continuous layer in this example, is activated when either modality is confident.

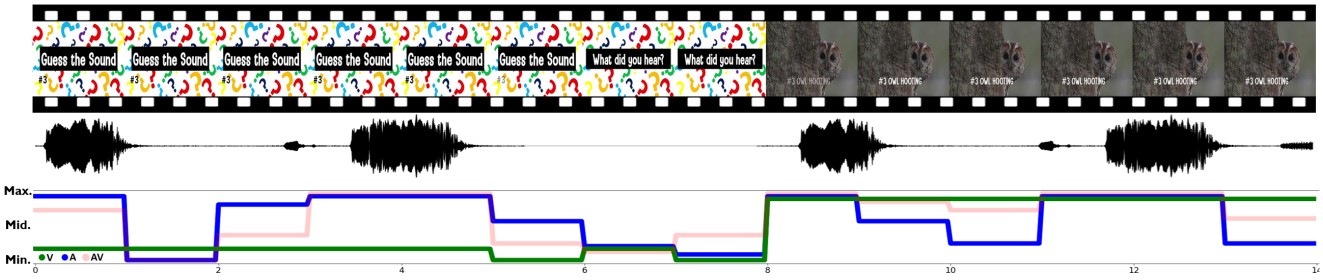

Figure 5: **Modality-level Video Analysis Results.** Plotted lines at the bottom of the figure depicts the modality confidence level throughout the video according to the informative signals.

F.3. Missing Label Detection

Often, a video contains multiple objects and events with complex interactions. It is difficult to fully represent contents of a video with a single label. Moreover, humans make mistakes during annotations by missing annotation. Our model outputs multiple labels using the event-specific layers which capture different characteristics of a video. We provide multi-label prediction examples using videos from VGGSound in Figure 6. The examples show that our model captures diverse contents of a video that are not annotated.

G. Annotation Process for VGGSound Multilabel Analysis (Multi-labeled VGGSound)

In the experiment section of the main paper, we perform multilabel analysis on VGGSound. Here, we describe the multilabel selection process and illustrate our tool for the analysis. Note that this is not a typical annotation process that finds all labels in a video. Since our human resource is limited [2], we focus on our analysis to answer two question: "Does the VGGSound dataset contain multiple events but only annotated with a single label?", and "Are the multi-label predictions of our network correct?'

[2]This multilabel user study is conducted with the research interns who are not involved in this work.

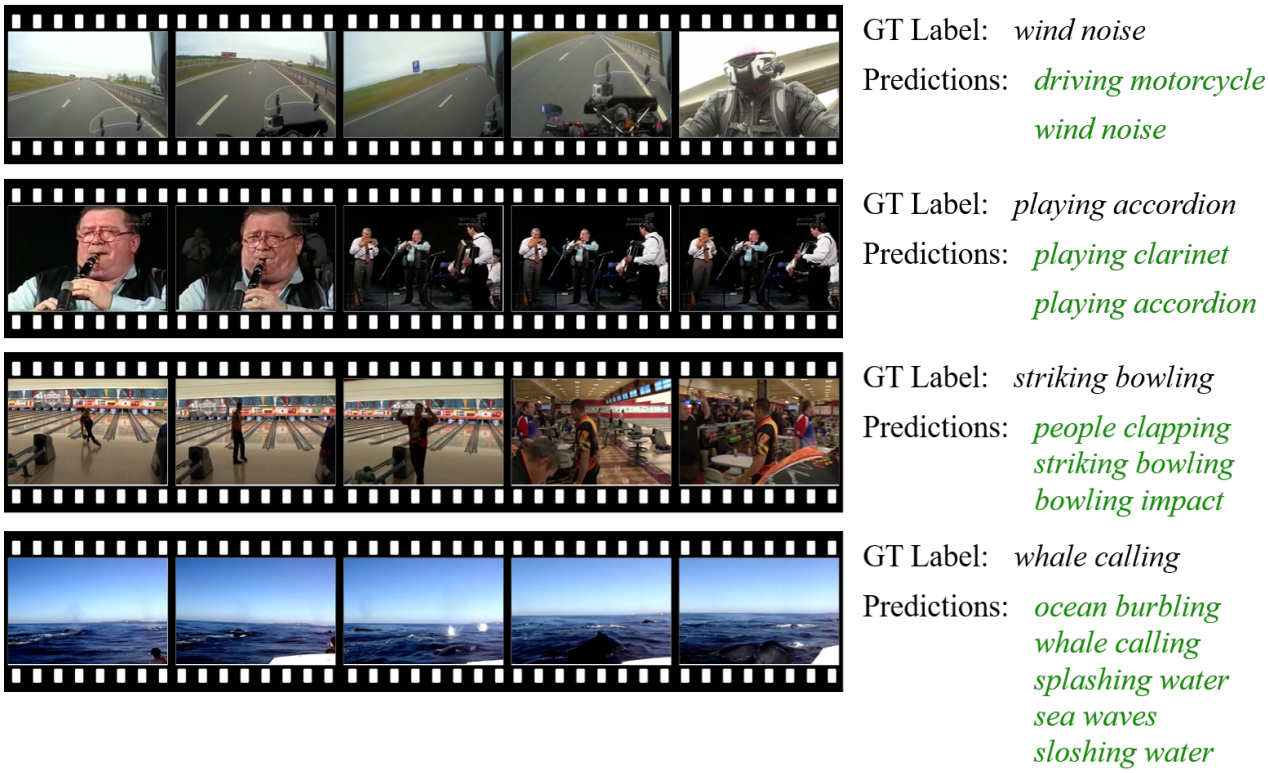

Figure 6: **Missing Label Prediction.** Our method predicts additional labels that are contained but not labeled in the given video examples.

As Figure 7 illustrates, our user interface to annotate multilabels on the VGGSound dataset is a web form with five columns. The first column, "Index", represents the index of the videos in order. The second column, "File Name", represents the name of the videos. The third column, "Video", contains the playable video (containing the audio as well). The fourth column, "Predictions", holds the potential class labels for the given video. The last column, "Category Info.", contains the supplemental visual description of each candidate class label. During the annotation process, annotators watch a video placed in the third column. There are multiple candidate classes predicted from our model for the given video in the fourth column. Annotators select relevant labels after watching the video. Additionally, annotators can refer to the last column, "Category Info.", as it provides visual examples to aid annotators who are not familiar with the meaning of the label. After making the selection for each video in this web form, users simply submit their answers via clicking the submit button.

We ask 12 subjects to annotate ~1200 videos in VGGSound given the multiple video-level predictions obtained by our model as candidates, and the subjects annotate matched labels with given video contents.

Limitations. Our system can predict multiple labels. The first limitation of our work is the raw predictions of multiple labels that are not calibrated. A prediction with the highest confidence is not necessarily more dominant or accurate than the other predictions. Proper calibration of the confidences may enable our model to select the most prominent event in a video. Another limitation is false positives in multi-label predictions. Although a majority of our model's multi-label predictions align with human perception as in Table 2 and Figure 4 of the main paper, there is still a gap.

References

[1] Triantafyllos Afouras, Andrew Owens, Joon Son Chung, and Andrew Zisserman. Self-supervised learning of audio-visual objects from video. In *European Conference on Computer Vision (ECCV)*, 2020.

[2] Relja Arandjelović and Andrew Zisserman. Look, listen and learn. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[3] Honglie Chen, Weidi Xie, Triantafyllos Afouras, Arsha Nagrani, Andrea Vedaldi, and Andrew Zisserman. Localizing visual sounds the hard way. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

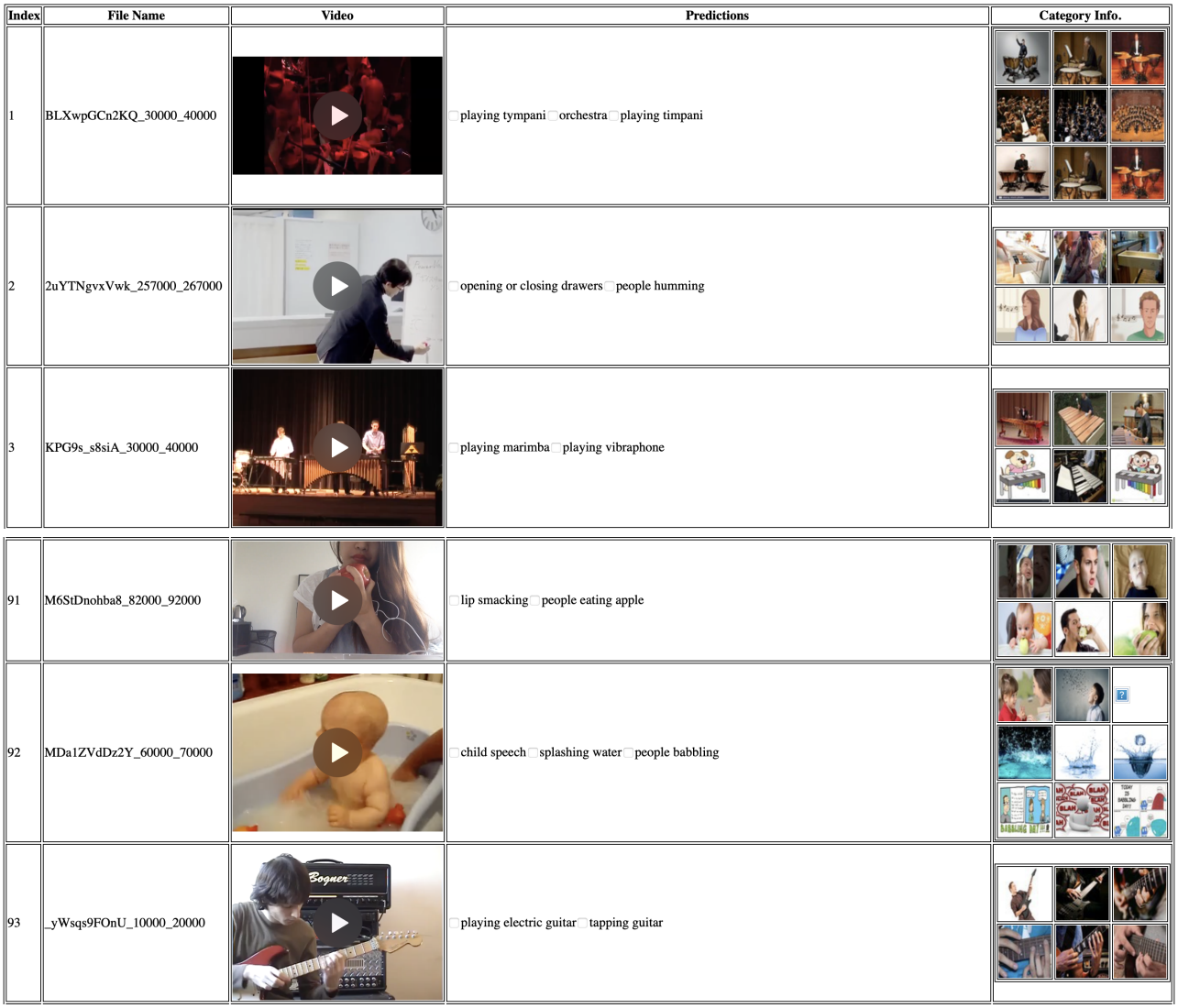

Figure 7: **User interface for multilabel analysis of VGGSound.**

[4] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020.

[5] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[6] 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.

[7] Arda Senocak, Tae-Hyun Oh, Junsik Kim, Ming-Hsuan Yang, and In So Kweon. Learning to localize sound source in visual scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[8] Zengjie Song, Yuxi Wang, Junsong Fan, Tieniu Tan, and Zhaoxiang Zhang. Self-supervised predictive learning: A negative-free method for sound source localization in visual scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[9] Yapeng Tian, Dingzeyu Li, and Chenliang Xu. Unified multisensory perception: Weakly-supervised audio-visual video parsing. In *European Conference on Computer Vision (ECCV)*, 2020.

[10] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *European Conference on Computer Vision (ECCV)*, 2018.

[11] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[12] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition. *arXiv preprint arXiv:2001.08740*, 2020.