AdaMML: Adaptive Multi-Modal Learning for Efficient Video Recognition (Supplementary Material)

Rameswar Panda[1,†], Chun-Fu (Richard) Chen[1,†], Quanfu Fan[1], Ximeng Sun[2],
Kate Saenko[1,2], Aude Oliva[1,3], Rogerio Feris[1]
†: Equal Contribution
[1]MIT-IBM Watson AI Lab, [2]Boston University, [3]MIT

Section	Content
A	Dataset Details
B	Implementation Details
C	Comparison with More Fusion Baselines
D	Discussion on RGB Difference
E	Qualitative Results
F	Runtime Analysis

Table 1: **Supplementary Material Overview**

A. Dataset Details

We evaluate the performance of our approach using four standard video datasets, namely ActivityNet-v1.3 [2], FCVID [4], Mini-Sports1M [5] and Kinetics-Sounds [1]. Below we provide more details on each of the dataset.

ActivityNet. We use the v1.3 split which consists of more than 648 hours of untrimmed videos from a total of 20K videos. Specifically, this dataset has 10,024 videos for training, 4926 videos for validation and 5044 videos for testing with an average duration of 117 seconds. It contains 200 different daily activities such as: walking the dog, long jump, and vacuuming floor. As in literature, we use the training videos to train our network, and the validation set for testing as labels in the testing set are withheld by the authors. The dataset is publicly available to download at http://activity-net.org/download.html.

FCVID. Fudan-Columbia Video Dataset (FCVID) contains total 91,223 Web videos annotated manually according to 239 categories (45,611 videos for training and 45,612 videos for testing). The categories cover a wide range of topics like social events, procedural events, objects, scenes, etc. that form in a hierarchy of 11 high-level groups (183 classes are related to events and 56 are objects, scenes, etc.). The total duration of FCVID is 4,232 hours with an average video duration of 167 seconds. The dataset is available to download at http://bigvid.fudan.edu.cn/FCVID/.

Mini-Sports1M. Mini-Sports1M is a subset of Sports-1M [5] dataset with 1.1M videos of 487 different fine-grained sports. It is assembled by [3] using videos of length 2-5 mins, and randomly sample 30 videos for each class for training, and 10 videos for each class for testing. The classes are arranged in a manually-curated taxonomy that contains internal nodes such as Aquatic Sports, Team Sports, Winter Sports, Ball Sports, etc, and generally becomes fine-grained by the leaf level. We obtain the training and testing splits from the authors of [3] to perform our experiments. Both training and testing videos in this dataset are untrimmed. This dataset is available to download at https://github.com/gtoderici/sports-1m-dataset.

Kinetics-Sounds. Kinetics-Sounds (assembled by [1]) is a subset of Kinetics and consists of 22,521 videos for training and 1,532 videos testing across 31 action classes. The original subset contains 34 classes, which have been chosen to be potentially manifested visually and aurally, such as playing various instruments (guitar, violin, xylophone, etc.), using tools (lawn mowing, shovelling snow, etc.), as well as performing miscellaneous actions (tap dancing, bowling, laughing, singing, blowing nose, etc.). Since 3 classes were removed from the original Kinetics dataset, we use the remaining 31 classes in our experiments, as in [3]. Although this dataset is fairly clean by construction, it still contains considerable noise and many videos contain sound tracks that are completely unrelated to the visual content (e.g. Doing fencing in Figure 3.(a) of the main paper) which makes it suitable for our approach to adaptively select right modalities conditioned on the input. The original Kinetics dataset is publicly available to download at https://deepmind.com/research/open-source/kinetics and the classes for Kinetics-Sounds can be obtained from [3].

B. Implementation Details

For our experiments, we use 12 NVIDIA Tesla V100 GPUs for the RGB + Audio experiments and 18 GPUs for both RGB + Flow and RGB + Flow + Audio experiments. All our models were implemented and trained via PyTorch.

Network. For non-audio modality, we add temporal max-pooling layers (kernel size 3, stride 2) to reduce computations. In recognition network, we use TSN-like ResNet-50 network [8] with three temporal max-pooling layers which are located at the beginning of stage 2, 3 and 4 of ResNet-50 (there are 4 stages in ResNet-50), i.e., the third, forth and fifth locations of reducing spatial resolution in the network. On the other hand, we add two temporal max-pooling layers to the MobileNetV2 used in the policy network for non-audio modality since the number input frames for policy network is fewer compared to the recognition network.

Input. We first use FFMPEG to extract RGB frames and Audio from a video. While decoding a video into RGB frames, the shorter side of the RGB frames is resized to 256 while keeping the aspect ratio. We use the resized frames to compute optical flow via TV-L1 algorithm and bound the flow range to [-20, 20]. We convert the audio to single-channel and resample it at 24kHz. During training, we divide a video into C equal-length regions ($C = 5$ in our experiments). For each region, we randomly pick 32 consecutive frames and uniformly subsample 8 frames as a RGB segment, i.e., the temporal stride between frames is 4. For the Audio data, we take a 1.28s-length window that is center-aligned to the RGB frames and then we use short-time Fourier transform to convert the audio into a log-spectrogram of window length 10ms, hop length 5ms with 256 frequency bins [6]. For the Flow data, at each RGB frame location, we stack horizontal and vertical flow of 5 consecutive frames interleavedly to form the input. Moreover, for the frame difference used in the multi-modal policy network, we follow the same practice as in optical flow, and stack 5 consecutive frame difference images to form the input [8]. On the other hand, C is 10 during testing as we use 10 video segments.

Training. We use a batch size of 72 with synchronized batch normalization in all our experiments, The data augmentations for the RGB and Flow modalities are based on the practices in [10]. We first randomly resize the shorter side of an image to a range of [256, 320) while keeping aspect ratio and then randomly crop a 224×224 region and normalize it with the ImageNet's mean and standard deviation to form the input ($8 \times 224 \times 224$). For the Audio modality, we simply take the 256×256 spectrogram as the input. The same data augmentations are used in the policy network while the data of the non-audio modality is further downsampled in both temporal and spatial dimension ($4 \times 160 \times 160$). The training time depends on the size of datasets and the task. E.g., for the RGB + Audio task, it takes about 12 hours for Kinetics-Sound and 16 hours for ActivityNet.

Testing. During testing, we uniformly sample 10 video segments from a video. For RGB and Flow modalities, we resize the shorter side of an image to 256, and then crop a center 224×224 region for evaluation.

Method	Acc. (%)	GFLOPs
RGB	82.85	141.36
Flow	75.73	163.39
RGBDiff	80.10	179.12
Weighted Fusion (RGB + Flow)	83.47	304.75
Weighted Fusion (RGB + RGBDiff)	83.30	320.48

Table 2: **Comparison between Optical Flow and RGB Difference on Kinetics-Sounds**. RGBDiff as an input modality is very competitive with optical flow in both unimodal and joint learning.

C. Comparison with More Fusion Baselines

Table 7 of the main paper shows that AdaMML outperforms five different fusion methods with $47\% - 55\%$ savings in GFLOPs on Kinetics-Sounds. We additionally compare with two mid-level fusion strategies (Unilateral and Bilateral connections as in [11]) and AdaMML still outperforms both by 1.6% and 1.2% in accuracy while requiring 55.9% and 56.8% less GFLOPs in RGB+Flow on Kinetics-Sounds. We also compare with one context gating baseline (collaborative experts where each modality is treated as an expert [7]) and AdaMML outperforms it by 4.87% in RGB+Flow on Kinetics-Sounds and 2.48% in RGB+Audio on ActivityNet. We also test gradient-blending [9] using author's released codes; however, it diverged during training on both Kinetics-Sounds and ActivityNet datasets.

D. Discussion on RGB Difference

As described in Section 4 of the main paper, we utilize RGB frame difference as a proxy to optical flow in our policy network and compute flow when needed since computing flow is very expensive. Here we compare RGBDiff and flow in terms of unimodal and weighted fusion (joint learning) when combined with RGB performance to further verify the effectiveness of RGBDiff on Kinetics-Sounds. Table 2 shows that RGBDiff outperforms Flow in unimodal performance (75.73% vs 80.10%) whereas both Flow and and RGBDiff (when combined with RGB) performs very similar (83.47 vs 83.30) on Kinetics-Sounds. This shows that RGBDiff as an input modality is also quite effective both in unimodal and joint learning performance and hence can be used as a proxy in the policy network for predicting the on-demand flow computation during test time.

E. Qualitative Results

Figure 1 shows the selected modalities using our approach on different cases. As seen from Figure 1.(a), our approach selects relevant RGB and audio for only first two segments as both modalities become irrelevant for last two segments as girls are discussing instead of cheerleading. Similarly, in Figure 1.(b), AdaMML is able to select RGB for only one segment that is more informative of the action and selects the entire audio stream as the action can be easily recognized

with audio (Playing Harmonica). Figure 1.(c) and (d) shows two more examples of RGB + Flow and RGB + Flow + Audio experiments respectively, where our approach selects the right modalities to use per segment (e.g., in Figure 1.(d), it mainly focuses on audio while selecting RGB and flow for only two segments) for correctly classifying the videos while taking efficiency into account.

F. Runtime Analysis

We compute runtime using an environment with PyTorch 1.2, CUDA 10.2, and a single NVIDIA Tesla V100 (32GB) GPU as our testbed. Our method still has advantages on actual inference speed. For instance, `AdaMML` delivers 1.96× (11.6 vs 5.9 videos/sec) and 1.42× (13.1 vs 9.22 videos/sec) speed up over the weighted fusion baseline that uses all the modalities irrespective of the input, on Kinetics-Sounds and ActivityNet datasets respectively.

References

[1] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *ICCV*, 2017. 1

[2] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 1

[3] Gao, Ruohan and Oh, Tae-Hyun, and Grauman, Kristen and Torresani, Lorenzo. Listen to look: Action recognition by previewing audio. In *CVPR*, 2020. 1

[4] Yu-Gang Jiang, Zuxuan Wu, Jun Wang, Xiangyang Xue, and Shih-Fu Chang. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *TPAMI*, 2017. 1

[5] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 1

[6] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *ICCV*, 2019. 2

[7] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts. *arXiv preprint arXiv:1907.13487*, 2019. 2

[8] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 2

[9] Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal networks hard? In *CVPR*, 2020. 2

[10] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018. 2

[11] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition. *arXiv preprint arXiv:2001.08740*, 2020. 2

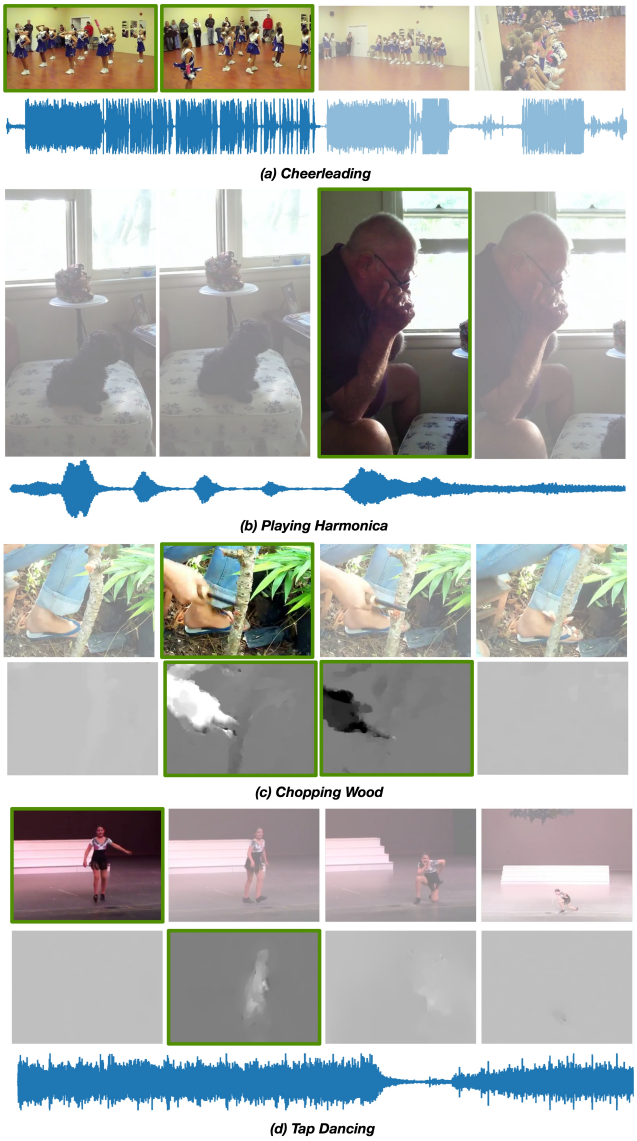

Figure 1: **More qualitative examples showing effectiveness of `AdaMML` in selecting right modalities per video segment (marked by green borders).** Overall, we observe that our approach focuses on the right modalities to use per segment for correctly classifying the videos while taking efficiency into account.