Towards Efficient Audio-Visual Learners via Empowering Pre-trained Vision Transformers with Cross-Modal Adaptation

Supplementary Material

6. Dataset Details

We evaluate our proposed STG-CMA on three different audio-visual understanding tasks including audio-visual event localization (AVE), audio-visual segmentation (AVS), and audio-visual question answering (AVQA). For all downstream tasks, we summarize the details of the used dataset separately in the following and present some descriptions in Table 8. In addition, we train our model on the training set and report the evaluation score on the testing set.

AVE dataset: We evaluate the performance of our STG-CMA on the AVE task using the AVE dataset [37] that is extracted from AudioSet. The AVE dataset consists of 4, 143 video clips, each one having a duration of 10 seconds and is labeled with corresponding events every 1 second. Meanwhile, the dataset includes 28 different event categories (*i.e.*, Dog Barking, Church Bell, Bus, Violin) and one background category without involving event information. Moreover, the AVE dataset is split into 3,339 video clips for training, 402 video clips for validation, and 402 video clips for testing.

AVSBench-S4 dataset: We validate our proposed STG-CMA on the AVSBench-S4 dataset [47] for the AVS task. The dataset contains 4, 932 video clips, where each one has a duration of 5 seconds. Meanwhile, the AVSBench-S4 dataset includes 23 categories including playing violin, baby laughter, lions roaring, etc. In addition, the pixel-level mask images are provided as the ground truth to represent the objects that generate the sound in the given RGB image frame. Moreover, the dataset is split into 3,452 videos for training, 740 videos for validation, and 740 videos for testing.

MUSIC-AVQA dataset: We conduct the experiments on the MUSIC-AVQA dataset [19] to evaluate the performance of our STG-CMA on the AVQA task. The dataset consists of 9, 288 video clips representing 45, 867 question-answer pairs including 33 question templates spanning over different modality scenarios (*i.e. Audio-Visual*, *Audio-only* and *Visual-only*) and question types (*i.e. Counting*, *Location*, *Comparative*, *Existential*, *Temporal*). In addition, the MUSIC-AVQA dataset has 42 types of answers for different question contents, such as '*Yes*' for existential questions, '*One*' for counting questions, '*Violin*' for temporal questions, etc.

Table 8. The description of used audio-visual datasets. Each dataset consists of raw video clips, which are then extracted into visual frames and audio waveforms as the audio-visual inputs. The 'Annotation Type' row presents whether the frames are annotated by category, pixel-level mask, or answer.

	AVE[37]	**AVSBench-S4** [47]	**MUSIC-AVQA** [19]
Video Clips	4, 143	4, 932	9, 288
Visual Frames	41, 430	24, 660	45, 867
Classes	29	23	42
Annotation Types	Event Category	Pixel-level Mask	Answer
Evalutation Score	Accuracy	mIoU	Accuracy

7. Implementation Details

In this section, we provide more implementation details about data pre-processing, model training and pre-trained vision backbone. We conduct all experiments with one NVIDIA GeForce RTX 3090 GPU by using the PyTorch framework.

7.1. Data Pre-processing

For visual input, following existing works [8, 25], our proposed STG-CMA receives 10 RGB image frames for AVE and AVQA tasks (M=10) and 5 RGB image frames for AVS (M=5), where the image frames are uniformly sampled from each video clip and each frame is then resized and cropped into the resolution of 224 × 224. For audio input, each audio waveform is chunked into 10 short segments for AVE and AVQA tasks (K=10) and 5 short segments for AVS task (M=5), where each one is converted into either 128-D (for CLIP-based backbone) or 224-D (for Swin-based backbone) fbank features by using Hanning window. In addition, for the AVE task, we follow [43] to adopt stronger data augmentation for visual signal (*i.e.*, random augmentation and random erasing), and follow [25] to use the mix-up augmentation for the audio signal.

7.2. Model Training

We train our proposed model in all audio-visual downstream tasks for 20 epochs by using the Adam [18] optimizer. Besides, we adopt the Cosine Decay [28] to dy-

Table 9. Training configurations and hyperparameters used for different audio-visual understanding tasks

Config	**AVE**		**AVS**	**AVQA**	
Model	STG-CMA (Tiny or Base)		STG-CMA (Base)	STG-CMA-B	STG-CMA-L
Backbone	CLIP-ViT	Swin-ViT	Swin-ViT	Swin-ViT	
Optimizer	Adam				
Adapter LR	5e-5		3e-4	5e-5	2.5e-5
Task-specific LR	5e-6		3e-4	5e-5	2.5e-5
Minimal LR	2e-6		2e-5	5e-6	2e-6
Weight Decay	5e-7				
Optimizer Momentim	(0.95, 0.999)				
Batch Size	1		2	2	
LR Schedule	Cosine Decay				
Warmup Epochs	2		5	2	
Loss Function	CE		IoU	CE	
Mixup	Yes		No	No	
Stronger Augmentation	Yes		No	No	

namically adjust the learning learning during the training procedure. More specifically, the learning rate is first linearly warmed up into initial value within the first several epochs and then decayed into the minimal one with a cosine function. To better train the model, we also assign different learning rates for updating parameters of newly introduced adapter layers and downstream layers for various audio-visual datasets. All training configurations or hyperparameters are summarized in Table 9.

7.3. Pre-trained Vision Transformers

We adopt the off-the-shelf pre-trained vision transformers ($i.e.$ CLIP and Swin transformers) as the backbone for both visual and audio encoders. The CLIP consists of vision and text transformer encoders, which are pre-trained on massive image-text pairs by using contrastive learning. We just employ the vision transformer encoder from CLIP as the frozen backbone in our method. In addition, the Swin transformer can efficiently produce the hierarchical feature representation while reducing the computational complexity using shift windows and cross-window attention, which is very useful for dense prediction tasks like audio-visual segmentation.