Supplementary Material for ECLIPSE: Efficient Long-range Video Retrieval using Sight and Sound

Our supplementary material consists of:

1. Implementation Details.
2. Additional Quantitative Results.
3. Additional Qualitative Results.
4. A Supplementary Video.

1 Implementation Details

Experimental Setting. In all experiments, the visual frames are extracted at 3 fps. We adopt pretrained CLIP [1] on both text and visual encoder, which is based on the ViT-B/32 visual backbone. We initialize the weights of our proposed audiovisual block using the corresponding spatial attention weights of CLIP. To gradually incorporate audio information into visual features, we attach a learnable fully connected layer to each audiovisual attention block and initially set it to zero. For visual representations, we first "patchify" 224×224 video frames into 32×32 patches as is done in [1]. Each video frame is then tokenized into 49 patches and a learnable 768-dimensional *CLS* token. At the end, the frame-level *CLS* tokens are averaged to obtain a video-level feature embedding that is used to optimize our model as described in Eq. (5),(6) of the main paper. For audio encoder, we use ResNet-18 [2] pre-trained on VGGSound [3]. We sample 10-second audio clips in the neighborhood around the sampled video frame and process the raw audio into spectrogram as is done in [3]. Lastly, for textual features, we adopt CLIP tokenizer for all text inputs. Specifically, the textual encoder processes all textual tokens and a special 768-dimensional *CLS* token as its inputs. Afterward, we only consider the *CLS* textual token to match a given video with the corresponding textual description.

Training Details. We implement ECLIPSE using Pytorch [4] and conduct the training on four NVIDIA A6000 GPUs. For fair comparison with the baseline methods, we set the batch size to 64. We train our model with Adam optimizer [5] and set the learning rate to $1e-7$ for text encoder and spatial attention in E.q (2) of main paper with weight decay 0.2. For our audiovisual attention blocks, A2V and V2A (see E.q (3) and E.q (4) in our main draft) , we set the learning rate to $1e-5$ with no weight decay. The maximum text input is set to 64 tokens for ActivityNet Captions, DiDeMo, and QVHighlight. We set 128 for YouCook2 due to longer paragraph.

Table 1. We investigate how different video frame sampling strategies affect the performance of a **video-only** CLIP4Clip [6] baseline on ActivityNet Captions [7]. The results are reported in text-to-video R@1 metrics. We observe that for a smaller number of frames (e.g., 32-64) random sampling yields slightly better performance than the uniform sapling. Conversely, for a larger number of frames (e.g., 96-128) uniform sampling leads to better accuracy.

Method \ Num. Frames	32	64	96	128
Uniform	40.4	40.7	**41.7**	40.9
Random Sample	41.0	41.2	40.9	40

2 Additional Quantitative Results

Ablating Different Frame Sampling Strategies. In Table 1, we investigate different video frame sampling strategies on ActivityNet Captions using R@1 evaluation metric. Specifically, we experiment with uniform and random frame sampling using a CLIP4Clip baseline [6]. For uniform sampling, we sample the frames uniformly throughout the entire input video. For random sampling, we divide the video into a fixed number of segments, and randomly sample one frame within each segment. Based on the results in Table 1, we note that random sampling improves performance for a smaller number of video frames (e.g., 32-64). Conversely, when using a larger number of frames (e.g., 96-128) the uniform sampling strategy leads to slightly better accuracy. For simplicity, we use the standard uniform sampling strategy for all of our experiments.

3 Additional Qualitative Results

Video Retrieval Results. In Figure 1, we provide additional qualitative results of our long-range video retrieval framework on ActivityNet Captions [7]. In all of these examples, we notice that CLIP4Clip baseline fails to capture relevant audio-based events (e.g., people cheering). In comparison, our EclipSE model successfully retrieves videos that contain complex audiovisual events, thus, highlighting the importance of audiovisual modeling for long-range video retrieval.

Sound Localization Results. In Figure 2, we also demonstrate qualitative sound localization results of our method. Specifically, by computing the similarity between audio features and visual patches, we can obtain saliency maps that are indicative of sound sources in the video. Furthermore, we would like to emphasize that our EclipSE model does not require any additional sound localization training objective. In other words, EclipSE successfully learns associations between sound sources and objects (e.g., a woman talking, a man playing the violin, a man using a chainsaw) as a byproduct of being trained for the video retrieval task.

4 Supplementary Video

Lastly, our supplementary material also includes a video *735.mp4* illustrating our qualitative results in the video format. Specifically, we include the results of our EclipSE model on several challenging video retrieval cases. For comparison, we also include the results of a CLIP4Clip baseline. Additionally, in these video results, we demonstrate that EclipSE also learns to localize sounds in the video even though it was not explicitly trained to do so. Overall, our video results indicate that compared to CLIP4Clip, EclipSE is more robust when retrieving long videos particularly in cases that involve complex audiovisual events.

References

1. Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1
2. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1
3. Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP*, 2020. 1
4. Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 1
5. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 1
6. Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. CLIP4Clip: An empirical study of clip for end to end video clip retrieval. *arXiv Preprint*, 2021. 2, 4
7. Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017. 2, 4

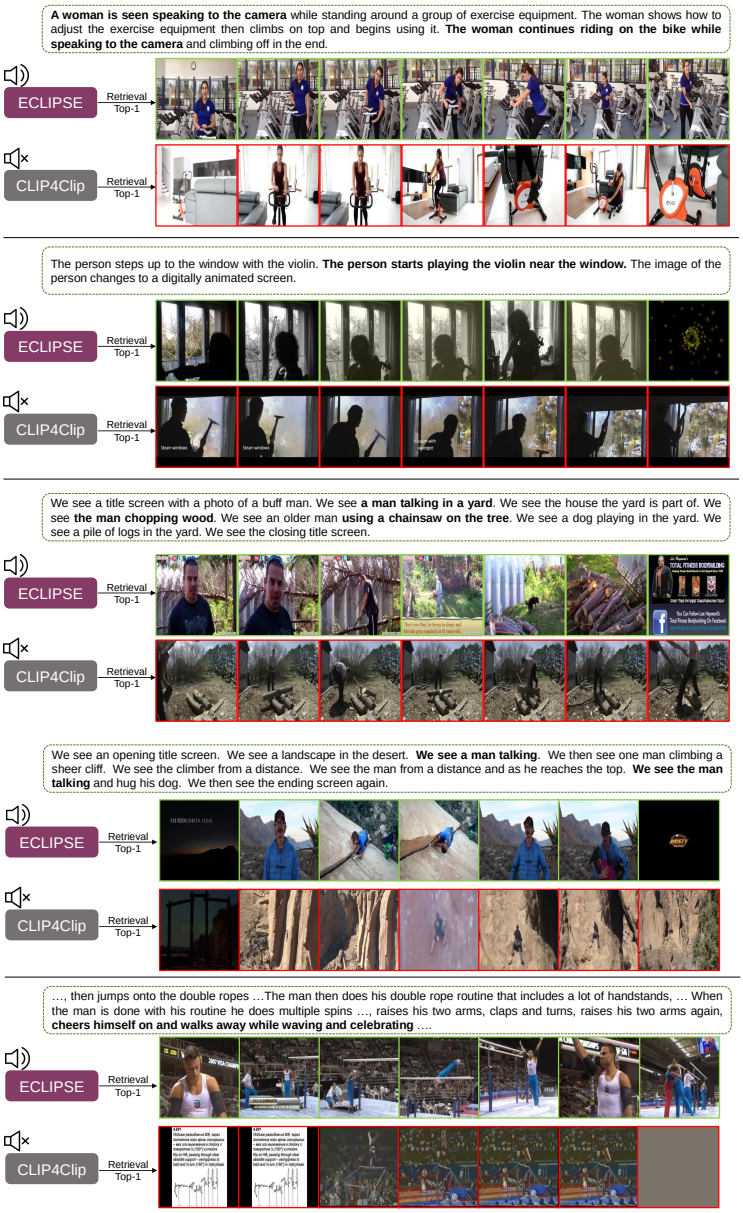

Fig. 1. Here, we illustrate our qualitative long-range retrieval results on ActivityNet Captions [7]. We compare our audiovisual EclipSE model with a video-only CLIP4Clip [6]. For a given a textual query (depicted in a green block), we visualize each method's top-1 retrieved video. Our results indicate that the video-only CLIP4Clip struggles with retrieval when textual queries include audio event descriptions, e.g., "a man talking", "a person cheering," etc. (see bolded text). In these cases, CLIP4Clip fails to retrieve the correct video instances, whereas EclipSE effectively leverages audiovisual cues for successful long video retrieval.

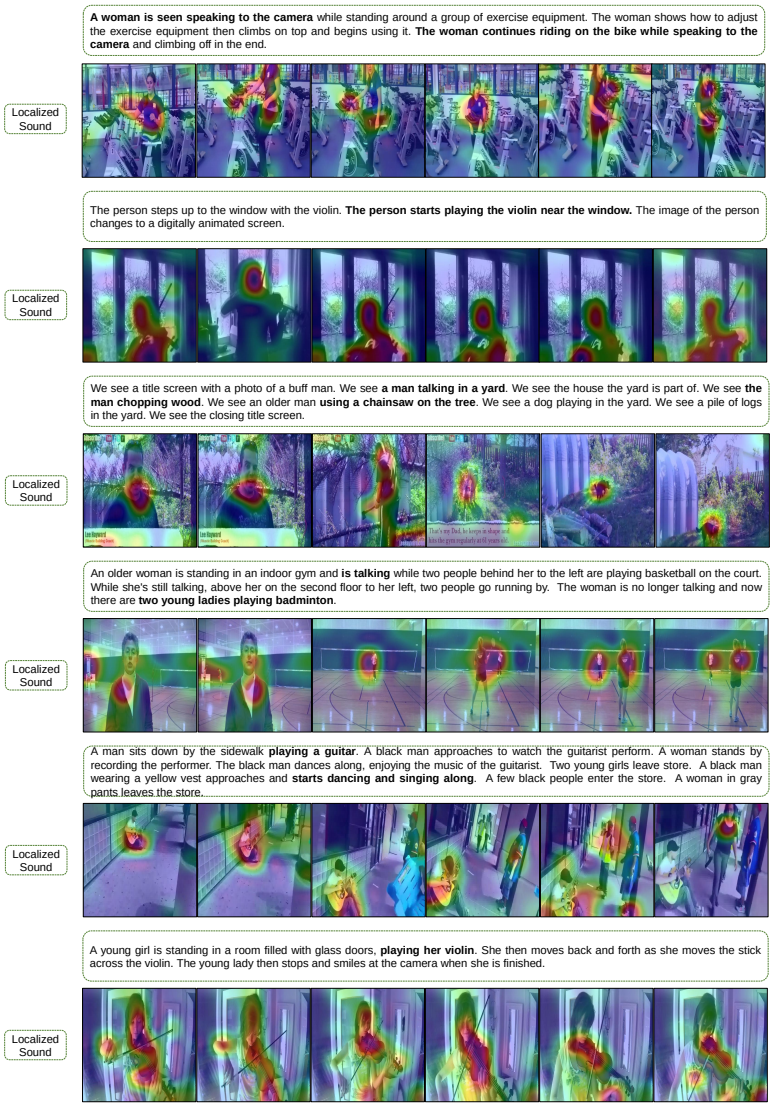

Fig. 2. Here, we illustrate qualitative sound localization results of our method. Note that our EclipSE is not explicitly trained for the sound localization task. In other words, EclipSE learns implicit associations between objects and sounds while being optimized with respect to the video retrieval task.