Supplementary Material:
Shot Contrastive Self-Supervised Learning for Scene Boundary Detection

Shixing Chen Xiaohan Nie* David Fan* Dongqing Zhang Vimal Bhat Raffay Hamid
Amazon Prime Video
{shixic, nxiaohan, fandavi, zdongqin, vimalb, raffay}@amazon.com

In this supplementary material, we: (a) present further details of our experimental settings for better reproducibility, (b) provide additional quantitative results for more comprehensive analysis, and (c) show additional qualitative results to provide a better understanding of our shot contrastive learning approach (ShotCoL).

1. Experiment Details

We used PyTorch [6] and Tesla V100 GPUs for all our experiments. For contrastive learning, we used 8 GPUs with the PyTorch module DistributedDataParallel. Below, we provide the network parameters, hyper-parameters and training details for the experiments on each dataset.

1.1. MovieNet Dataset

This section corresponds to §4.2.2 in the main paper.

1.1.1 Training Details

a. Contrastive Learning: Recall that we used ResNet-50 [2] with the first layer modified to take 9 input channels as our visual encoder. The weights of the query encoder θ_q were randomly initialized. The weights of the key encoder θ_k were initialized to the same values as θ_q. The query encoder was trained using SGD with a mini-batch size of 256, momentum of 0.9, and weight decay of 0.0001. The initial learning rate of 0.03 was dropped twice each time by $10\times$.

b. Supervised Learning: For training the multi-layer perceptron (MLP) for the scene boundary detection task on the MovieNet [3] dataset, we used dropout value of 0.9, batch size of 1024, maximum epoch number of 200 and SGD with a fixed learning rate of 0.1.

1.1.2 Additional Results

a. Visual Modality: We present the top 1 accuracy, top 5 accuracy and loss curves during contrastive training in Figure 1. There are two types of weight initialization methods compared: (i) randomly initialized, and (ii) pre-trained on ImageNet [1] dataset. Note that if the network was pre-trained on ImageNet [1] dataset before contrastive learning, the loss for the first few epochs are relatively low, but the converged loss is quite close to the randomly initialized network. This shows that our method does not rely on pre-trained weights, and is capable of learning from scratch.

We can see that once training performance saturates and the curves flatten, *e.g.*, at epoch 60, the performance can be further improved by decaying the learning rate. We decay the learning rate twice at epochs 60 and 90 for the randomly initialized network, and at epochs 30 and 60 for the network

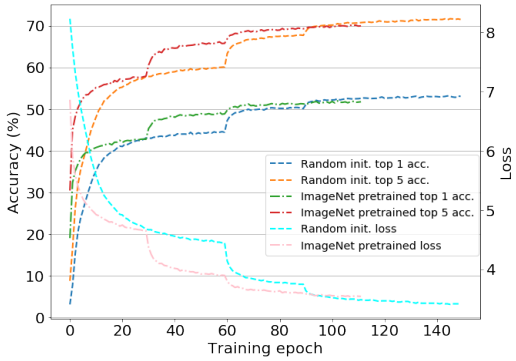

Figure 1: Training curves of contrastive learning on MovieNet dataset.

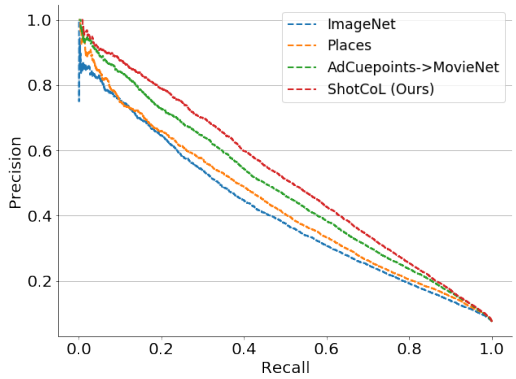

Figure 2: PR-curves of test set in MovieNet dataset.

*Equal contribution.

pre-trained on ImageNet.

To provide a more intuitive understanding of the AP results in Table 2 of the main paper, we show in Figure 2 the Precision Recall (PR)-curves on MovieNet test set after performing supervised learning on MovieNet training set.

b. Audio Modality: As the video files in the MovieNet dataset [3] are not yet released, we only had access to the keyframes and the pre-computed audio features for each shot in the dataset. Concatenating our 1 keyframe-based visual shot-features with the provided pre-computed audio shot-features for scene boundary detection resulted in only marginal AP improvement from 52.34 to 52.47. This is in contrast to our results on the AdCuepoints dataset where incorporating learned audio features along with the learned visual features resulted in substantive AP improvement (see Table 5 of the main paper). This observation highlights the importance of using raw audio to learn audio features for scene boundary detection, and suggests that having access to raw audio for MovieNet data could further improve the results of our approach on MovieNet dataset.

1.1.3 Positive Key Selection

During contrastive learning, we used ImageNet space to select our initial set of positive keys while our encoder weights were randomly initialized. This allowed us to exploit the underlying film-production process and select the right set of neighborhood shots that could offer informative data augmentation required to learn an effective embedding space.

Since the computational cost of updating the positive key set is high, we only updated the positive key set occasionally (at epochs 20 and 50) during training. Each update can be viewed as a re-initialization step such that between consecutive re-initializations, the argmax operation used to select positive keys does not impact differentiability.

To further distill the importance of using ImageNet space in particular for initial positive key set selection, we compare how positive key sets evolve over the course of training when they are initially selected using: (a) ImageNet space, versus (b) randomly generated space. Results from this comparison are given in Table 1, and explained below.

Let F_{imgnet} denote the feature-space learned using initial positive key-set obtained from ImageNet space, and K_{imgnet} denote the positive key-set at each training epoch. Similarly, use F_{random} to denote the feature-space of randomly initialized encoder, and K_{random} as the set of positive keys at each training epoch. Table A shows the percent overlap between K_{imgnet} and K_{random} at the end of different epochs. We make two key observations for the results in Table A.

First, even at epoch 0, the overlap between K_{imgnet} and K_{random} is already 32.12%, which is significantly larger than random chance (recall that our context is 16 shots long, making the probability of there being an overlap between the two sets by random chance to be $1/16 = 6.25\%$).

epoch #	0	20	40	60	80	100
overlap %	32.12	70.24	74.42	76.83	75.83	75.11

Table 1: Overlap between K_{imgnet} and K_{random} for different epochs.

Second, as training goes on the overlap between K_{imgnet} and K_{random} converges to $\sim 75\%$ by 100 epochs. Using this learned space for scene boundary detection task based on one keyframe produces an AP of 51.53% on the MovieNet dataset.

These observations lead us to believe that our approach stays stable so long as the feature-space used to find initial positive key-set is good enough. While ImageNet space worked out well for us, it is not a crucial requirement for the stability of our approach. In fact, even the space from randomly initialized encoder can work for our approach.

1.2. AdCuepoints

This section corresponds to §4.3.2 of the main paper.

1.2.1 Visual Modality

a. Contrastive Learning: For the visual modality of AdCuepoints dataset, we used the same settings as mentioned above in § 1.1.1-a.

b. Supervised Learning: For training the MLP for the ad cuepoint detection task using the visual modality of the AdCuepoints data, we used a fixed learning rate of 1.0, dropout value of 0.8, batch size of 1024, and maximum epoch number of 200.

1.2.2 Audio Modality

a. Contrastive Learning: We used the Wavegram-Logmel-CNN14 variant of PANNs [4] as our audio encoder. The architecture of this encoder is similar to VGG [7] but adapted for audio. The network uses a combined wavegram and log-mel spectrogram representation. The wavegram representation is learned by expanding the time-domain input waveform to include a third dimension and convolving over this expanded input. This extra axis is analogous to frequency and allows the network to learn a joint time-frequency representation.

During contrastive learning, the weights θ_q of query encoder were randomly initialized, and the weights θ_k of key encoder were initialized to the same values as θ_q. The query encoder was trained using Adam optimizer with a mini-batch size of 128, betas of 0.9 and 0.999, epsilon of $1e-08$, and no weight decay. The learning rate was initialized to 0.0005 and decayed using cosine annealing schedule[4].

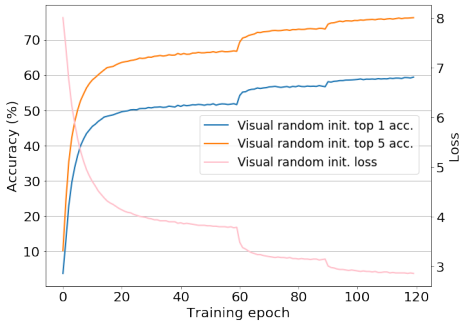

Figure 3: Training curves of visual contrastive learning on AdCuepoints dataset.

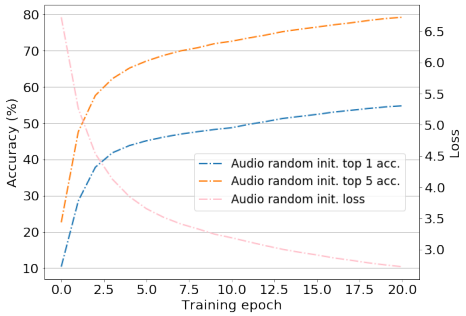

Figure 4: Training curves of audio contrastive learning on AdCuepoints dataset.

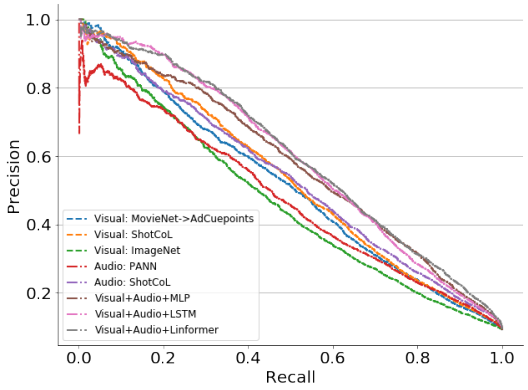

Figure 5: PR-curves of test set on AdCuepoints dataset.

b. Supervised Learning: For training the MLP for ad cuepoint detection task on the AdCuepoints dataset for audio modality, we used a fixed learning rate of 0.01, batch size of 512, no dropout and maximum epoch number of 100.

1.2.3 Audio-Visual Fusion

In Table 5 of the main paper, we discussed the use of more sophisticated temporal models, *i.e.* Bi-LSTM and Transformer, to jointly incorporate the visual and audio features. To do this, we added an FC layer before the temporal models to map the 4096-dimensional feature vector (audio + visual) to a more compact 2048-dimensional feature vector. The features were then passed through the temporal model, followed by MLP for prediction. For the experiments given in Table 5 of the main paper, each shot in the sample was treated as a separate time-step, so the sequence length was equivalent to the number of shots.

The Bi-LSTM was implemented using the LSTM module in PyTorch [6]. It had two layers with 2048 hidden units and dropout of 0.2 between them. We used Linformer [8] as our choice of Transformer, which implemented sparse self-attention with linear complexity, allowing much faster runtime and lower memory usage. There were 4 attention layers with 8 attention heads in each layer in the Linformer, and the projection dimension was 256. We prepended a classification token to the beginning of each sequence, and the final hidden state of this token was used as the sequence representation for downstream classification. The input features were reshaped to $batch\_size \times num\_shots \times feature\_len$, to treat features from adjacent shots in the input as separate timesteps before being passed to the Linformer. The output of Linformer is then flattened to $batch\_size \times num\_shots \cdot feature\_len$ to concatenate features from adjacent shots before being passed to the MLP. The same process is followed for the Bi-LSTM in order to treat features from adjacent shots as separate timesteps.

1.2.4 Additional Results

The training curves using the visual and audio modalities for contrastive learning on the AdCuepoints dataset are presented in Figure 3 and Figure 4 respectively. Note that the audio network converges notably faster than the visual network, which is consistent with the observations in [9]. In our case, this effect is exaggerated due to the use of Adam optimizer when training the audio encoder.

To provide a more intuitive understanding of the AP results in Table 3, 4, and 5 of the main paper, we show in Figure 5 the PR-curves on AdCuepoints test set after performing supervised learning on AdCuepoints training set.

Recall that for the visual modality of the MovieNet, we showed the effectiveness of using our proposed shot-similarity compared to existing image augmentation schemes (Table 2, row 8 and row 11 in the main paper). Along similar lines, we explored the effectiveness of using existing audio augmentation schemes (*e.g.* SpecAugment [5]) compared to our proposed audio-shot similarity for the AdCuepoints dataset. Contrasting a query shot with its augmented version (using SpecAugment [5]) during contrastive learning, we can achieve an AP of 50.53 using the 10 shot setting. Our shot similarity-learning approach instead can achieve an AP of 53.27. This result further validates applicability of our approach on audio modality.

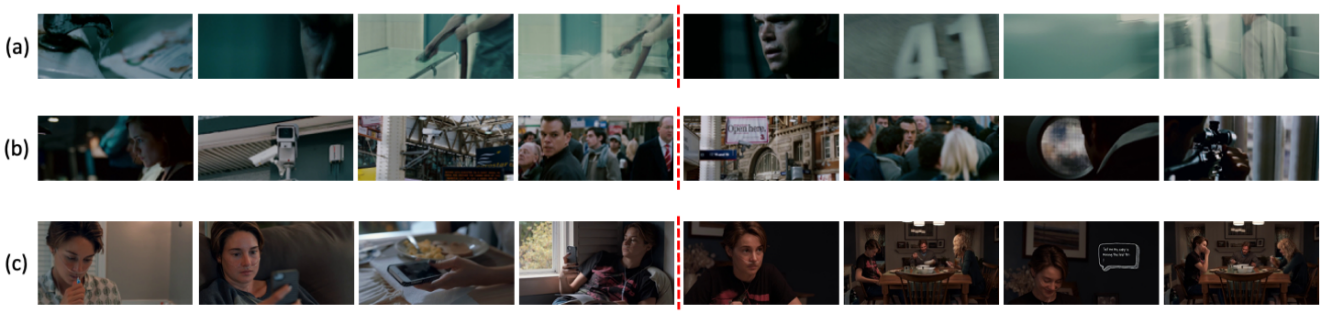

Figure 6: Examples in the test set of MovieNet. Labeled scene boundaries are shown with red dashed lines.

Figure 7: Labeled scene boundaries in MovieNet are shown with red dashed lines.

Figure 8: Labeled ad cuepoints in AdCuepoints are shown with red dashed lines.

2. Qualitative Results

2.1. Challenging Examples

To give an intuitive sense of the difficulty level of the scene boundary detection task, we show in Figure 6 a few examples of the labeled scene boundaries in the test set of MovieNet data. As can be observed, even humans can have difficulty confidently disambiguating whether the shot boundaries in Figure 6 are scene boundaries or not.

2.2. MovieNet vs AdCuepoints

Recall that ad cuepoints are a special case of scene boundaries. To provide more intuition behind this point, in Figure 7 and Figure 8 we present some representative examples that demonstrate the differences between the MovieNet and AdCuepoints datasets. As can be observed in Figure 7, scene boundaries can be semantically quite close to each other, and arbitrarily inserting video ads to such scene boundaries can break the flow of the storyline and interrupt the viewing experience of the audience.

In contrast, as shown in Figure 8 ad cuepoints are more distinguishable and isolated from each other. Therefore, inserting ads at such points is likely to result in minimal disruption of the storyline as the scenes before and after the ad cuepoints are more distinguishable and semantically unrelated from each other.

2.3. Additional Nearest Neighbor Results

Recall that we discussed the effectiveness of our learned shot representation for the task of scene boundary detection in §4.1 in the main paper. In Figure 9 of this document, we provide some additional results to underscore this point. We present 5 nearest neighbor shots retrieved for a query shot using different shot-representations. We compare our learned representation with ImageNet feature [1] and Places feature [10] to tell whether the 5 nearest neighbor shots are from the same scene or not. As can be observed, while results retrieved using Places and ImageNet features are visually quite similar to the query-shot, almost none of them are from the query-shot's scene (indicated on the top left of shot-frame). In contrast, results from our shot-representation are all from the same scene even though the appearance of the retrieved shots does not exactly match query shot. This shows that our learned shot-representation is able to effectively encode the local scene-structure.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1, 5

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1

[3] Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *The European Conference on Computer Vision (ECCV)*, 2020. 1, 2

[4] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894, 2020. 2

[5] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *Proc. Interspeech 2019*, pages 2613–2617, 2019. 3

[6] 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 *Advances in neural information processing systems*, pages 8026–8037, 2019. 1, 3

[7] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[8] Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. 3

[9] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition. *arXiv:2001.08740*, 2020. 3

[10] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 5

Figure 9: Additional nearest neighbor results. Shot indices are displayed at top-left corners where blue indicates query shot, green indicates shot from the same scene as query, and red indicates shot from a different scene.