Shot Contrastive Self-Supervised Learning for Scene Boundary Detection

Shixing Chen   Xiaohan Nie*   David Fan*   Dongqing Zhang   Vimal Bhat   Raffay Hamid

Amazon Prime Video

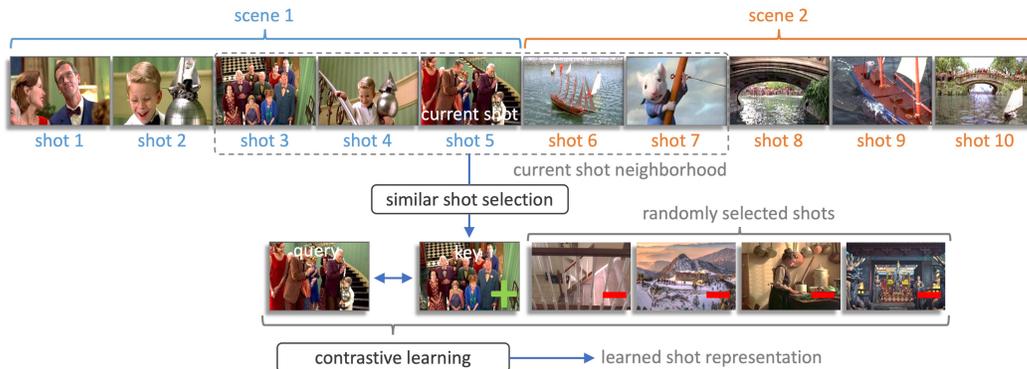{shixic, nxiaohan, fandavi, zdongqin, vimalb, raffay}@amazon.com

Figure 1: **Approach Overview** – Representative frames of 10 shots from 2 different scenes of the movie Stuart Little are shown. The story-arch of each scene is distinguishable and semantically coherent. We consider similar nearby shots (*e.g.* 5 and 3) as augmented versions of each other. This augmentation approach is able to capitalize on the underlying film-production process and can encode the scene-structure better than the existing augmentation methods. Given a current shot (query) we find a similar shot (key) within its neighborhood and: (a) maximize the similarity between the query and the key, and (b) minimize the similarity of the query with randomly selected shots.

Abstract

Scenes play a crucial role in breaking the storyline of movies and TV episodes into semantically cohesive parts. However, given their complex temporal structure, finding scene boundaries can be a challenging task requiring large amounts of labeled training data. To address this challenge, we present a self-supervised shot contrastive learning approach (ShotCoL) to learn a shot representation that maximizes the similarity between nearby shots compared to randomly selected shots. We show how to apply our learned shot representation for the task of scene boundary detection to offer state-of-the-art performance on the MovieNet [33] dataset while requiring only ~25% of the training labels, using 9× fewer model parameters and offering 7× faster runtime. To assess the effectiveness of ShotCoL on novel applications of scene boundary detection, we take on the problem of finding timestamps in movies and TV episodes where video-ads can be inserted while offering a minimally disruptive viewing experience. To this end, we collected a new dataset called AdCuepoints with 3,975 movies and TV episodes, 2.2 million shots and 19,119 minimally disruptive ad cue-point labels. We present a thorough empirical analysis on this dataset demonstrating the effectiveness of ShotCoL for ad cue-points detection.

*Equal contribution.

1. Introduction

In filmmaking and video production, shots and scenes play a crucial role in effectively communicating a storyline by dividing it into easily interpretable parts. A **shot** is defined as a series of frames captured from the same camera over an uninterrupted period of time [40], while a **scene** is defined as a series of shots depicting a semantically cohesive part of a story [23] (see Figure 1 for an illustration). Localizing shots and scenes is an important step towards building semantic understanding of movies and TV episodes, and offers a broad range of applications including preview generation for browsing and discovery, content-driven video search, and minimally disruptive video-ads insertion.

Unlike shots which can be accurately localized using low-level visual cues [38] [6], scenes in movies and TV episodes tend to have complex temporal structure of their constituent shots and therefore pose a significantly more difficult challenge for their accurate localization. Existing unsupervised approaches for scene boundary detection [3] [34] [2] do not offer competitive levels of accuracy, while supervised approaches [33] require large amounts of labeled training data and therefore do not scale well. Recently, several self-supervised learning approaches have been applied to learn generalized visual representations for images [22] [1] [16] [18] [28] [48] [51] [43] and short video

clips [32] [12] [46] [41], however it has been mostly unclear how to extend these approaches to long-form videos. This is primarily because the relatively simple data augmentation schemes used by previous self-supervised methods cannot encode the complex temporal scene-structure often found in long-form movies and TV-episodes.

To address this challenge, we propose a novel shot contrastive learning approach (ShotCoL) that naturally makes use of the underlying production process of long-form videos where directors and editors carefully arrange different shots and scenes to communicate the story in a smooth and believable manner. This underlying process gives rise to a simple yet effective invariance, *i.e.*, nearby shots tend to have the same set of actors enacting a semantically cohesive story-arch, and are therefore in expectation more similar to each other than a set of randomly selected shots. This invariance enables us to consider nearby shots as augmented versions of each other where the augmentation function can implicitly capture the local scene-structure significantly better than the previously used augmentation schemes. Specifically, given a shot, we try to: (a) maximize its similarity with its most similar neighboring shot, and (b) minimize its similarity with a set of randomly selected shots (see Figure 1 for an illustration).

We show how to use our learned shot representation for the task of scene boundary detection to achieve state-of-the-art results on MovieNet dataset [33] while requiring only ∼25% of the training labels, using $9\times$ fewer model parameters, and offering $7\times$ faster runtime. Besides these performance benefits, our single-model based approach is significantly easier to maintain in a production setting compared to previous approaches that make use of multiple models [33].

As a practical application of scene boundary detection, we explore the problem of finding timestamps in movies and TV episodes for minimally disruptive video-ads insertion. To this end, we present a new dataset called AdCuepoints with $3,975$ movies and TV episodes, 2.2 million shots, and $19,119$ manually labeled minimally disruptive ad cue-points. We present a thorough empirical analysis on this dataset demonstrating the generalizability of ShotCoL on the task of ad cue-points detection.

2. Related Work

Self-Supervised Representation Learning: Self supervised learning (SSL) is a class of algorithms that attempts to learn data representations using unlabeled data by solving a surrogate (or *pretext*) task using supervised learning. Here the supervision signal for training can be automatically created [22] without requiring labeled data. Some of the previous SSL approaches have used the pretext task of reconstructing artificially corrupted inputs [45] [30] [49], while others have tried classifying inputs into a set of pre-defined categories with pseudo-labels [8] [9] [29].

Contrastive Learning: As an important subset of SSL methods, contrastive learning algorithms attempt to learn data representations by contrasting similar data against dissimilar data while using contrastive loss functions [27]. Contrastive learning has shown promise for multiple recognition based tasks for images [16] [22] [4]. Recently, with a queue-based mechanism that enables the use of large and consistent dictionaries in a contrastive learning setting, the momentum contrastive approach [14] [5] has demonstrated significant accuracy improvement compared to the earlier approaches. Recent works on using contrastive learning for video analysis [32] [11][44][21] primarily focus on short-form videos where relatively simple data augmentation approaches have been applied to learn the pretext task. In contrast, our work focuses on long-form movies and TV episodes where we learn shot representations by incorporating a data augmentation mechanism that can exploit the underlying filmmaking process and therefore can encode the local scene-structure more effectively.

Scene Boundary Detection: Scene boundary detection is the problem of identifying the locations in videos where different scenes begin and end. Earlier methods for scene boundary detection such as [37], adopt an unsupervised-learning approach that clusters the neighboring shots into scenes using spatiotemporal video features. Similar to [37], the work in [34] clusters shots based on their color similarity to identify potential boundaries, followed by a shot merging algorithm to avoid over-segmentation. More recently, supervised learning approaches [36] [2] [31] [33] have been proposed to learn scene boundary detection using human-annotated labels. While these approaches offer better accuracy compared to earlier unsupervised approaches, they require large amounts of labeled training data and are therefore difficult to scale.

Multiple datasets have been used to evaluate scene boundary detection approaches. For instance, the OVSD dataset [36] includes 21 videos with scene labels and scene boundaries. Similarly, the BBC planet earth dataset [2] consists of 11 documentaries labeled with scene boundaries. Recently, the MovieNet dataset [19] has taken a major step in this direction and published $1,100$ movies where 318 of them are annotated with scene boundaries. Building on this effort to scale up the evaluation for scene boundary detection and its applications, we present empirical results on a new dataset called AdCuepoints with $3,975$ movies and TV episodes, 2.2 million shots, and $19,119$ manual labels.

3. Method

We first discuss our self-supervised approach for shot-level representation learning where we present the details of our encoder network and contrastive learning approach. We
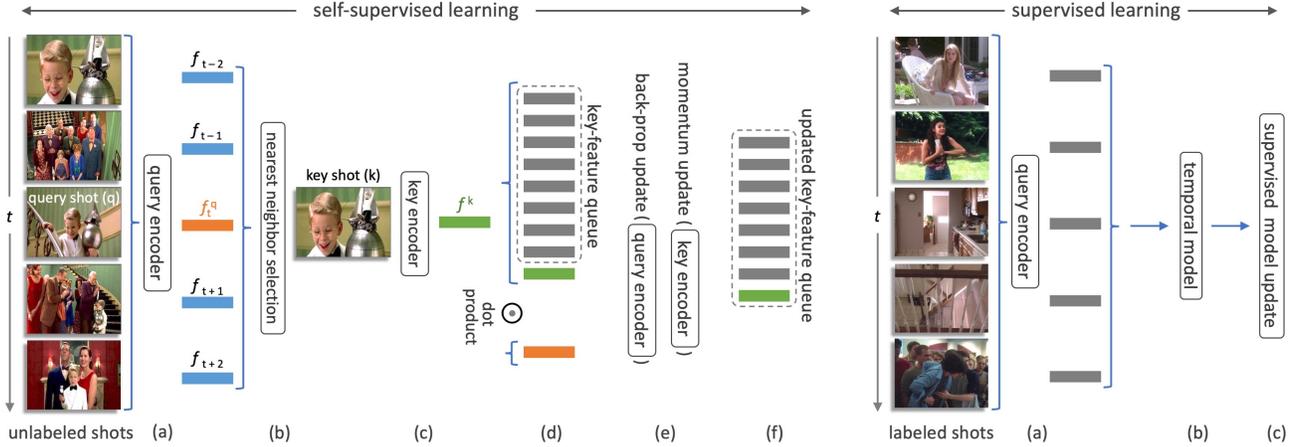
Figure 2: **Self-Supervised Learning**: (a) Use unlabeled data to extract the **visual or audio** features of a given query shot and its neighboring shots. (b) Find the key shot which is most similar to the query shot within its neighborhood. (c) Pass the key shot through the key encoder. (d) Contrast the query shot feature with key shot feature and the set of already queued features. (e) Use a contrastive loss function to update the query encoder through back-propagation and use momentum update for the key encoder. (f) Insert the key shot feature into the key-feature queue. **Supervised Learning**: (a) Use labeled data to extract visual or audio features of all shots by using the query encoder trained during the self-supervised learning step. (b) Learn temporal information among the shots. (c) Update the network using supervised learning.

then discuss how we use our trained encoder in a supervised learning setting for the task of scene boundary detection. Our overall approach is illustrated in Figure 2.

3.1. Shot-Level Representation Learning

Given a full-length input video, we first use standard shot detection techniques [38] to divide it into its constituent set of shots. Our approach for shot representation learning has two main components: (a) encoder network for visual and audio modalities, and (b) momentum contrastive learning [14] to contrast the similarity of the embedded shots. We now present the details for these two components.

3.1.1 Shot Encoder Network

We use separate encoder networks to independently learn representations for the audio and visual modalities of the input shots. Although ShotCoL is amenable to using any encoder network, the particular encoders we used in this work incorporate simplifications that are particularly conducive to scene boundary detection. The details of our visual and audio encoder networks are provided below.

1- Visual Modality: Since scene boundaries exclusively depend on inter-shot relationships, encoding intra-shot frame-dynamics is not as important to us. We therefore begin by constructing a 4D tensor (w, h, c, k) from each shot with uniformly sampled k frames each with w width, h height and c color channels. We then reshape this 4D tensor into a 3D tensor by combining the c and k dimensions together. This conversion offers two key advantages:

a. Usage of Standard Networks: As multiple standard networks (*e.g.* AlexNet [26], VGG [39], ResNet [35], *etc.*) support 2D images as input, by considering shots as 3D tensors we are able to directly apply a wide set of standard image classification networks for our problem.

b. Resource Efficiency: As we do not keep the time dimension explicitly after the first layer of our encoder network, we require less memory and compute resources compared to using temporal networks (*e.g.* 3D CNN [13]).

Specifically, we use ResNet-50 [15] as our encoder for the visual modality which produces a 2048-dimensional feature vector to encode the visual signal for each shot.

2- Audio Modality: To extract the audio embedding from each shot, we use a Wavegram-Logmel CNN [25] which incorporates a 14-layer CNN similar in architecture to the VGG [17] network. We sample 10-second mono audio samples at a rate of 32 kHz from each shot. For shots that are less than 10 seconds long, we equally zero-pad the left and right to form a 10-second sample. For shots longer than 10 seconds, we extract a 10-second window from the center. These inputs are provided to the Wavegram-Logmel network [25] to extract a 2048-dimensional feature vector for each shot.

3.1.2 Shot Contrastive Learning

We apply contrastive learning [10] to obtain a shot representation that can effectively encode the local scene-structure and is therefore conducive for scene boundary detection. To

| query shot | previous | next | augmented | ShotCoL |

Figure 3: Different ways to select positive key given a query shot.

this end, we propose to use a *pretext*[1] task that is able to exploit the underlying film-production process and encode the scene-structure better than the recent alternative video representations [32] [12] [46] [41].

For a given *query* shot, we first find the positive *key* as its most similar shot within a neighborhood around the query, and then: (a) maximize the similarity between the query and the positive key, and (b) minimize the similarity of the query with a set of randomly selected shots (*i.e.* negative keys). For this pretext task no human annotated labels are used. Instead, training is entirely based on the *pseudo-labels* created when the pairs of query and key are formed.

a. Similarity and Neighborhood: More concretely, for a query at time t denoted as q_t, we find its positive key k_0 as the most similar shot in a neighborhood consisting of $2 \times m$ shots centered at q_t. This similarity is calculated based on the embeddings of the query encoder $f(\cdot|\theta_q)$:

$$k_0 = \arg \max_{x \in X_t} f(q_t|\theta_q) \cdot f(x|\theta_q) \qquad (1)$$

$$X_t = [q_{t-m}, ..., q_{t-2}, q_{t-1}, q_{t+1}, q_{t+2}, ..., q_{t+m}] \qquad (2)$$

Along with K negative keys S_K, the K+1 shots ($k_0 \cup S_K$) are encoded by a key encoder to form a (K+1)-class classification task, where q needs to be classified to class k_0.

Our pretext task can be considered as training an encoder for a dictionary look-up task [14], where given a query, the corresponding key should be matched. In our case, given an input query shot q, the goal is to find its positive key shot k_0 in a set of shots $\{k_0, k_1, k_2, \ldots, k_K\}$. By defining the similarity as a dot product, we use the contrastive loss function InfoNCE [28]:

$$\mathcal{L}_q = -\log \frac{\exp(f(q|\theta_q) \cdot g(k_0|\theta_k)/\tau)}{\sum\limits_{i=0}^{K} \exp(f(q|\theta_q) \cdot g(k_i|\theta_k)/\tau)} \qquad (3)$$

where $g(\cdot|\theta_k)$ is the key encoder with the parameter θ_k. Here k_0 is the positive key shot, and $k_1, k_2, \ldots, k_K$ are negative key shots. Also, τ is the temperature term [48] such that when $\tau = 1$, Equation 3 becomes standard log-loss function with softmax activation for multi-class classification.

The intuition behind our method of positive key selection is illustrated in Figure 3, where given a query shot, different ways to select its positive key are shown. Notice that using

image-focused augmentation schemes (col. 4) as done in *e.g.* [14] does not incorporate any information about scene-structure. Similarly, choosing a shot adjacent to the query shot (col. 2 and 3) as the key can result in a too large and unrelated appearance difference between the query and key. Instead, selecting a similar nearby shot as the positive key provides useful information related to the scene-structure and therefore facilitates learning a useful shot representation. Results showing the ability of our shot representation to encode scene-structure are provided in § 4.1.

b. Momentum Contrast: Although large dictionaries tend to lead to more accurate representations, they also incur additional computational cost. To address this challenge, [14] recently proposed a queue-based solution to enable large-size dictionary training. Along similar lines, we save the embedded keys in a fixed-sized queue as negative keys. When a new mini-batch of keys come in, it is enqueued, and the oldest batch in the queue is dequeued. This allows the computed keys in the dictionary to be re-used across mini-batches.

To ensure consistency of keys when the key encoder evolves across mini-batch updates, a momentum update scheme [14] is used for the key encoder, with the following update equation:

$$\theta_k \leftarrow \alpha \cdot \theta_k + (1 - \alpha) \cdot \theta_q \qquad (4)$$

where α is the momentum coefficient. As only θ_q is updated during back-propagation, θ_k can be considered as a moving average of θ_q across back-propagation steps.

3.2. Supervised Learning

Recall that scenes are composed of a series of contiguous shots. Therefore, we formulate the problem of scene boundary detection as a binary classification problem of determining if a shot boundary is also a scene boundary or not.

To this end, after dividing a full length video into its constituent shots using low-level visual cues [38], for each shot boundary we consider its $2 \times N$ neighboring shots (N shots before and N shots after the shot boundary) as a data-point to perform scene boundary detection.

For each data-point, we use the query encoder trained by contrastive learning to extract shot-level visual or audio features independently. We then concatenate the feature vectors of the $2 \times N$ shots into a single vector, which is then provided as an input to a multi-layer perceptron (MLP) classifier[2]. The MLP consists of three fully-connected (FC) layers where the final FC layer is followed by softmax for normalizing the logits from FC layers into class probabilities of the positive and negative classes. Unless otherwise mentioned, the weights of the trained encoder are kept fixed during this step, and only MLP weights are learned.

---

[1]We use the terms *pretext*, *query*, *key* and *pseudo-labels* as their standard usage in contrastive learning literature. See [14] for more information.

[2]Note that other classifiers besides MLP can also be used here. See § 5 for comparative results of using different temporal models.
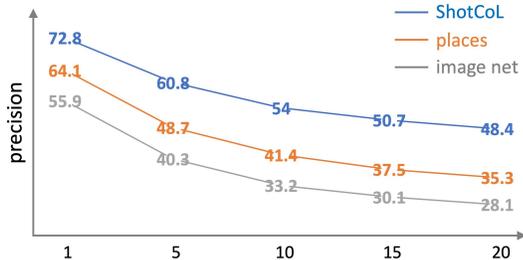
Figure 4: Comparison of shot retrieval precision (y-axis) using the test split of the MovieNet dataset [19] with different number of nearest neighbors (x-axis) and shot representations.

During inference, for each shot boundary, we form the $2 \times$ N-shot sample, extract shot feature vectors and pass the concatenated feature to our trained MLP to predict if the shot boundary is a scene boundary or not.

4. Experiments

We first present results to distill the effectiveness of our learned shot representation in terms of its ability to encode the local scene-structure, and then use detailed comparative results to show its competence for the task of scene boundary detection. Finally, we demonstrate the results of Shot-CoL for a novel application of scene boundary detection, *i.e.* finding minimally disruptive ad cue-points.

4.1. Effectiveness of Learned Shot Representation

Intuitively, if a shot representation is able to project shots from the same scenes to be close to each other, it should be useful for scene boundary detection. To test how well our learned shot representation is able to do this, given a query shot from a movie, we retrieve its k nearest neighbor shots from the same movie. Retrieved shots belonging to the same scene as the query shot are counted as true positives, while those from other scenes as false positives. We use the test split of MovieNet [19], and compare our learned shot representation (see § 4.2 for details) with Places [50] and ImageNet [7] features computed using ResNet-50 [15].

Results in Figure 4 show that our learned shot representation significantly outperforms other representations for a wide range of neighborhood sizes, demonstrating its ability to encode the local scene-structure more effectively.

Figure 5 provides an example qualitative result where 5 nearest neighbor shots for a query shot using different shot representations are shown. While results retrieved using Places [50] and ImageNet [7] features are visually quite similar to the query shot, almost none of them are from the query shot's scene. In contrast, results from ShotCoL representation are all from the same scene even though the appearances of the retrieved shots do not exactly match query shot. This shows that our learned shot representation is able to effectively encode the local scene-structure.

Figure 5: Five nearest neighbor shots for a query shot using different shot representations are shown. Shot indices are displayed at top-left corners where green indicates shot from same scene as query while red indicates shot from a different scene.

4.2. Scene Boundary Detection

We now present comparative performance of various models for scene boundary detection using MovieNet data [19].
a. Evaluation Metrics: We use the commonly used metrics to evaluate the considered methods [33], *i.e.* Average Precision (AP), Recall and Recall@3s, where Recall@3s calculates the percentage of the ground truth scene boundaries which are within 3 seconds of the predicted ones.

b. Dataset: Our comparative analysis for scene boundary detection uses the MovieNet dataset [19] which has $1,100$ movies out of which 318 have scene boundary annotations. The train, validation and test splits for these 318 movies are already provided by authors of MovieNet [19] with 190, 64 and 64 movies respectively. The scene boundaries are annotated at shot level and three key frames are provided for each shot. Following [33], we report the metrics on only the test set for all of our experiments unless otherwise specified.

c. Implementation Details: We use all $1,100$ movies with $\sim$1.59 million shots in MovieNet [19] to learn our shot representation, and 190 movies with scene boundary annotations to train our MLP classifier. All weights in the encoder and MLP are randomly initialized. For contrastive learning settings, as 80% of all scenes in MovieNet are 16 shots or less, we fix the neighborhood size for positive key selection to 8 shots. Other hyper-parameters are similar to MoCo [14], *i.e.*, $65,536$ queue size, 0.999 MoCo momentum, and 0.07 softmax temperature. The initial set of positive keys is selected based on the space of ImageNet (details in Supplementary Material). We use a three-layer MLP classifier (number-of-shots-used$\times$2048-4096-1024-2), and use dropout after each of the first two FC layers.

4.2.1 Ablation Study

Focusing on visual modality, we evaluate ShotCoL on the validation set of MovieNet for: (a) different number of shots, and (b) different number of key frames used per shot.

# of keyframes	# of shots				
	2	4	6	8	10
1	48.66	55.24	54.89	53.89	52.94
3	48.95	**56.13**	55.73	54.01	53.07

Table 1: AP results for ablation study in MovieNet data [19].

As shown in Table 1, using 2 shots in ShotCoL does not perform well signifying that the context within 2 shots is not enough for classifying scene boundaries accurately. The features using 4 shots achieve the highest AP, however the AP decreases when more shots are included. This is because as the context becomes larger, there is a higher chance of having multiple scene boundaries in each sample which makes the task more challenging for the model. In terms of the number of keyframes, the shot representation learned using 3 keyframes performs better than the one using only 1 keyframe. This indicates that the subtle temporal relationship within each shot can be beneficial for distinguishing different scenes.

Based on this ablation study, for all our experiments we use 3 frames per shot. For all of our scene boundary detection experiments we use a context of 4 shots (two to the left and two to the right) around each shot transition point to form a positive or negative sample based on its label.

4.2.2 Comparative Empirical Analysis

The detailed comparative results are given in Table 2. LGSS [33] has been the state-of-the-art on the MovieNet data [19] reporting 47.1 AP achieved by using four pretrained models (two ResNet-50, one ResNet-101 and one VGG-m) on multiple modalities together with LSTM. We comfortably outperform LGSS [33] (relative margin of 13.3%) using a single network on visual modality only. Moreover, ShotCoL offers 9× fewer model parameters and 7× faster runtime compared to LGSS [33].

Recall that results in [33] were reported using 150 titles from MovieNet [19] with 100, 20 and 30 titles for training, validation and testing respectively. Therefore, we also provide results on the 150 titles subset of MovieNet [19] (called MovieScenes [33]). As the exact data-splits are not provided by [33], we do a 10-fold cross-validation and report the mean and standard deviation, showing 12.1% relative performance gain over [33] in expectation.

To compare our shot contrastive learning with previous self-supervised methods, we focus on two recently proposed methods outlined in [14] and [4]. For each of these approaches, we consider two types of data augmentation strategies: (a) traditional image augmentation schemes (as used in [14] and [4]), and (b) our proposed shot augmentation scheme. Results in Table 2 show that using image-focused augmentation schemes only marginally improves the performance over the ImageNet baseline. In contrast,
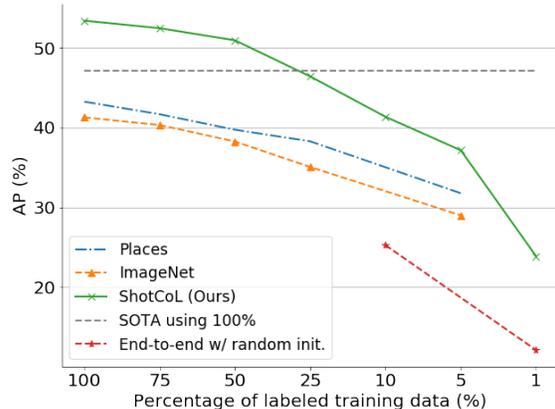

Figure 6: Results on different label-amounts for MovieNet [19] data. Dashed Gray line is for LGSS [33] with 100% labels.

using our proposed shot augmentation scheme with either [14] or [4] results in significant improvements.

Limited Amount of Labeled Training Data: The comparative performance of using our learned shot representation in limited labeled settings is given in Figure 6. Our learned feature is able to achieve 47.1 test AP (results reported by LGSS [33]) while using only ∼25% of training labels.

Moreover, we compare the performance of ShotCoL with an end-to-end learning based setting with limited labeled data following the protocols in [4]. As shown in Figure 6, learning an end-to-end model with random initialization and limited training labels is challenging. Instead, ShotCoL is able to achieve significantly better performance using limited number of training labels.

4.3. Application – Ad Cue-Points Detection

To assess the effectiveness of ShotCoL on novel applications of scene boundary detection, we take on the problem of finding timestamps in movies and TV episodes where video-ads can be inserted while being minimally disruptive. Such timestamps are referred to as *ad cue-points*, and are required to follow multiple constraints. First, cue-points can only occur when the context of the storyline clearly and unambiguously changes. This is different from scene boundaries observed in other datasets such as MovieNet [19], where the before and after parts of a scene can be contextually closely related. Second, cue-points cannot have dialogical activity in their immediate neighborhood. Third, all cue-points must be a certain duration apart and their total number needs to be within a certain limit which is a function of the video length. These constraints make ad cue-points detection a special case of scene boundary detection.

4.3.1 AdCuepoints Dataset

The AdCuepoints dataset contains 3,975 movies and TV episodes, 2.2 million shots, and 19,119 manually labeled

	Models	Modalities	Est. # of Encoder Parameters	Est. inference time/batch	AP	Recall (0.5 thr.)	Recall@3s (0.5 thr.)
			Without self-supervised pre-training				
1	SCSA [3]	Visual	23 m	6.6s	14.7	54.9	58.0
2	Story Graph [42]	Visual	23 m	6.6s	25.1	58.4	59.7
3	Siamese [2]	Visual	23 m	6.6s	28.1	60.1	61.2
4	ImageNet [7]	Visual	23 m	2.64s	41.26	30.06	33.68
5	Places [50]	Visual	23 m	2.64s	43.23	59.34	64.62
6	LGSS [33]	Visual Audio Action Actor	228 m	39.6s	47.1	73.6	79.8
			With self-supervised pre-training				
7	SimCLR [4] (img. aug.)	Visual	23 m	2.64s	41.65	75.01	80.42
8	MoCo [14] (img. aug.)	Visual	23 m	2.64s	42.51	71.53	77.11
9	SimCLR [4] (shot similarity)	Visual	25 m	5.39s	50.45	81.31	**85.91**
10	ShotCoL (MovieScenes [33])	Visual	25 m	5.39s	52.83 ±2.08	81.59 ±1.82	85.44 ±1.46
11	ShotCoL	Visual	25 m	5.39s	**53.37**	**81.33**	85.34

Table 2: **Comparative analysis for scene boundary detection –** The compared methods are grouped in two, *i.e.*: (a) ones that do not use self-supervised learning and (b) ones that use self-supervised learning followed by use of learned features in a supervised setting.
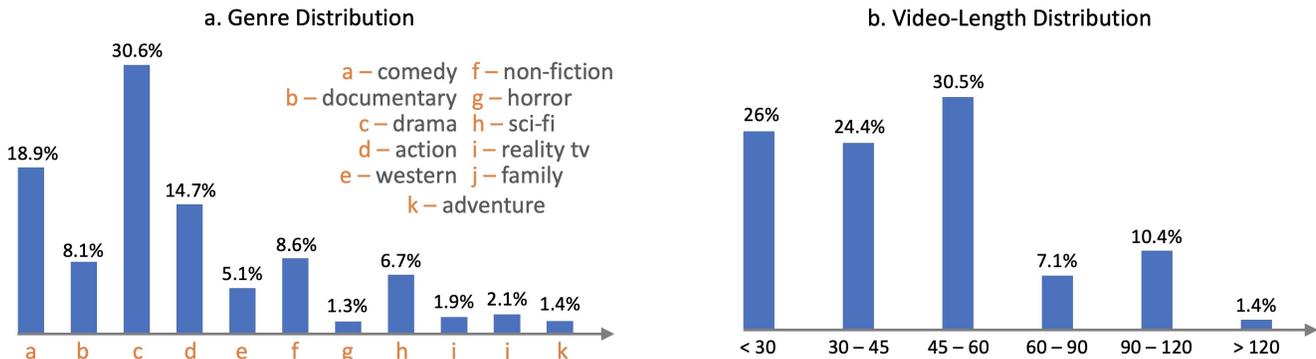

Figure 7: **a –** Distribution of video genres in AdCuepoints dataset. **b –** Distribution of video length in AdCuepoints dataset.

cue-points. Compared to the MovieNet dataset [19] which only contains movies, the AdCuepoints dataset also contains TV episodes which makes it more versatile and diverse from a content-based perspective. The video distribution of various genres present in AdCuepoints dataset is given in Figure 7-a. The distribution of video lengths in the AdCue-points dataset is provided in Figure 7-b.

We divide the $3,975$ full-length videos in the AdCue-points dataset into their constituent shots by applying commonly used shot detection approaches [38]. Recall that cue-points always lie at shot boundaries. We consider k shots to the left and right of each cue-point to create a positive sample with $\pm k$ context. Negative samples are created around positive samples by taking a sliding-window traversal to the left and right of positive samples while incorporating a unit stride. We divide our dataset into training, validation and testing sets with 70%, 10%, 20% ratio respectively.

4.3.2 Results

a. Visual Modality: We learn our shot representation using the entire unlabeled AdCuepoints dataset, and then apply it along with other representations as inputs to MLP models that use cue-point labels for training. Table 3 shows that our shot representation performs significantly better than the alternatives. Here ImageNet [7] features on 2D-CNN [15] and Kinetics [24] features on 3D-CNN [13] provide baselines.

Note that even when using the shot similarity features self-trained on unlabeled MovieNet data [19], the results

Visual Feature		AP
Pre-training data	Labeled data	
1 ImageNet [7]	AdCuepoints	45.90
2 Kinetics [24]	AdCuepoints	46.33
3 AdCuepoints	AdCuepoints	**53.98**
4 MovieNet	AdCuepoints	51.32
5 AdCuepoints	MovieNet	48.40

Table 3: Performance of using different visual features on Ad-Cuepoints dataset and cross-dataset results.

Audio	# of shots				
Feature	2	4	6	8	10
PANN [25]	43.56	46.47	47.17	46.97	47.40
ShotCoL	49.38	52.56	52.7	53.45	53.27

Table 4: Performance comparison of using pre-trained audio features [25] with ShotCoL based audio feature.

	MLP	B-LSTM [20] + MLP	Linformer [47] + MLP
# of shots	4	10	10
# of parameters	71 m	197 m	190 m
AP	57.65	59.02	59.95

Table 5: Comparison with different temporal models on the combined audio-visual feature.

obtained on AdCuepoints test data are significantly better than baseline. Similar trend can be observed on the cross-dataset setting of training our shot representation on unlabeled AdCuepoints data and testing on the MovieNet data [19], where we achieve 48.40% AP. These results demonstrate that our learned shot representation is able to generalize well in a cross-dataset setting.

b. Audio Modality: Following the aforementioned procedure of our visual modality comparison, Table 4 presents the results of using pre-trained audio features [25] compared with audio features learned using ShotCoL on Ad-Cuepoints dataset. Results using different number of shots are presented showing that ShotCoL is able to outperform existing approach by a sizable margin, demonstrating its effectiveness on audio modality.

c. Audio-Visual Fusion: Table 5 shows how combining learned audio and visual features can help further improve the performance of ShotCoL. Column 1 shows the result for concatenating our learned audio and visual shot representations and providing them as input to an MLP model. Moreover, columns 2 and 3 demonstrate that incorporating more sophisticated temporal models (*i.e.* B-LSTM [20] and Linformer[47]) can help fuse the audio and visual modalities more effectively than using simple feature concatenation. This shows that our shot representation can be used with a broad class of models downstream.
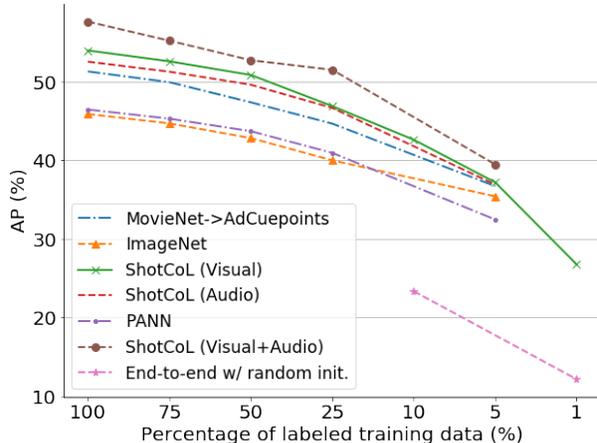

Figure 8: Comparative performance when using different amounts of training data for AdCuepoints dataset.

d. Limited Amount of Labeled Data: The comparison of different shot representations when using limited amounts of labeled training data is provided in Figure 8. It can be observed that ShotCoL is able to comfortably outperform all other considered methods on the AdCuepoints dataset. Moreover, we compare ShotCoL with an end-to-end learning setting as [4], where we use only 10% and 1% of the labeled training data. It can be seen that using our learned features with limited labeled data is able to give significantly better performance compared to using end-to-end learning.

5. Conclusions and Future Work

We presented a self-supervised learning approach to learn a shot representation for long-form videos using unlabeled video data. Our approach is based on the key observation that nearby shots in movies and TV episodes tend to have the same set of actors enacting a cohesive story-arch, and are therefore in expectation more similar to each other than a set of randomly selected shots. We used this observation to consider nearby similar shots as augmented versions of each other and demonstrated that when used in a contrastive learning setting, this augmentation scheme can encode the scene-structure more effectively than existing augmentation schemes that are primarily geared towards images and short videos. We presented detailed comparative results to demonstrate the effectiveness of our learned shot representation for scene boundary detection. To test our approach on a novel application of scene boundary detection, we take on automatically finding ad cue-points in movies and TV episodes and use a newly collected large-scale data to show the competence of our method for this application.

Going forward, we will focus on improving the efficiency of contrastive video representation learning. We will also investigate the application of our shot representation to additional problems in video understanding.

References

[1] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, 2019.

[2] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. A deep siamese network for scene detection in broadcast videos. In *Proceedings of the 23rd ACM International Conference on Multimedia*, 2015.

[3] Vasileios T Chasanis, Aristidis C Likas, and Nikolaos P Galatsanos. Scene detection in videos using shot clustering and sequence alignment. *IEEE Transactions on Multimedia*, 2008.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[6] Costas Cotsaces, Nikos Nikolaidis, and Ioannis Pitas. Video shot detection and condensed representation. a review. *IEEE Signal Processing Magazine*, 2006.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[8] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[9] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2014.

[10] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[11] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. In *European Conference on Computer Vision*, 2020.

[12] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *Advances in Neural Information Processing Systems*, 2020.

[13] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[15] 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*, 2016.

[16] Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[17] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017.

[18] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019.

[19] Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *European Conference on Computer Vision*, 2020.

[20] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[21] Dinesh Jayaraman and Kristen Grauman. Slow and steady feature analysis: Higher order temporal coherence in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[22] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[23] Ephraim Katz. The film encyclopedia. *Thomas Y. Crowell*, 1979.

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

[25] 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. *arXiv preprint arXiv:1912.10211*, 2020.

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[27] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. Contrastive representation learning: A framework and review. *IEEE Access*, 2020.

[28] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[29] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[30] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[31] Stanislav Protasov, Adil Mehmood Khan, Konstantin Sozykin, and Muhammad Ahmad. Using deep features for video scene detection and annotation. *Signal, Image and Video Processing*, 2018.

[32] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. *arXiv preprint arXiv:2008.03800*, 2020.

[33] Anyi Rao, Linning Xu, Yu Xiong, Guodong Xu, Qingqiu Huang, Bolei Zhou, and Dahua Lin. A local-to-global approach to multi-modal movie scene segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[34] Zeeshan Rasheed and Mubarak Shah. Scene detection in hollywood movies and tv shows. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.

[36] Daniel Rotman, Dror Porat, and Gal Ashour. Optimal sequential grouping for robust video scene detection using multiple modalities. *International Journal of Semantic Computing*, 2017.

[37] Yong Rui, Thomas S Huang, and Sharad Mehrotra. Exploring video structure beyond the shots. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, 1998.

[38] Panagiotis Sidiropoulos, Vasileios Mezaris, Ioannis Kompatsiaris, Hugo Meinedo, Miguel Bugalho, and Isabel Trancoso. Temporal video segmentation to scenes using high-level audiovisual features. *IEEE Transactions on Circuits and Systems for Video Technology*, 2011.

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

[40] Robert Sklar. Film: An international history of the medium. *Thames and Hudson*, 1990.

[41] Li Tao, Xueting Wang, and Toshihiko Yamasaki. Self-supervised video representation using pretext-contrastive learning. *arXiv preprint arXiv:2010.15464*, 2020.

[42] Makarand Tapaswi, Martin Bauml, and Rainer Stiefelhagen. Storygraphs: visualizing character interactions as a timeline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[43] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European Conference on Computer Vision*, 2020.

[44] Michael Tschannen, Josip Djolonga, Marvin Ritter, Aravindh Mahendran, Neil Houlsby, Sylvain Gelly, and Mario Lucic. Self-supervised learning of video-induced visual invariances. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[45] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, 2008.

[46] Jinpeng Wang, Yuting Gao, Ke Li, Xinyang Jiang, Xiaowei Guo, Rongrong Ji, and Xing Sun. Enhancing unsupervised video representation learning by decoupling the scene and the motion. *arXiv preprint arXiv:2009.05757*, 2020.

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

[48] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[49] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision*, 2016.

[50] 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*, 2017.

[51] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.