Movies2Scenes: Using Movie Metadata to Learn Scene Representation

Shixing Chen Chun-Hao Liu Xiang Hao Xiaohan Nie Maxim Arap Raffay Hamid
Amazon Prime Video
{shixic, chunhaol, xianghao, nxiaohan, maxarap, raffay}@amazon.com

Abstract

Understanding scenes in movies is crucial for a variety of applications such as video moderation, search, and recommendation. However, labeling individual scenes is a time-consuming process. In contrast, movie level metadata (e.g., genre, synopsis, etc.) regularly gets produced as part of the film production process, and is therefore significantly more commonly available. In this work, we propose a novel contrastive learning approach that uses movie metadata to learn a general-purpose scene representation. Specifically, we use movie metadata to define a measure of movie similarity, and use it during contrastive learning to limit our search for positive scene-pairs to only the movies that are considered similar to each other. Our learned scene representation consistently outperforms existing state-of-the-art methods on a diverse set of tasks evaluated using multiple benchmark datasets. Notably, our learned representation offers an average improvement of 7.9% *on the seven classification tasks and* 9.7% *improvement on the two regression tasks in LVU dataset. Furthermore, using a newly collected movie dataset, we present comparative results of our scene representation on a set of video moderation tasks to demonstrate its generalizability on previously less explored tasks.*

1. Introduction

Automatic understanding of movie scenes is a challenging problem [53] [26] that offers a variety of downstream applications including video moderation, search, and recommendation. However, the long-form nature of movies makes labeling of their scenes a laborious process, which limits the effectiveness of traditional end-to-end supervised learning methods for tasks related to automatic scene understanding.

The general problem of learning from limited labels has been explored from multiple perspectives [51], among which contrastive learning [28] has emerged as a particularly promising direction. Specifically, using natural language supervision to guide contrastive learning [41] has shown impressive results specially for zero-shot image-classification tasks. However, these methods rely on image-text pairs which are hard to collect for long-form videos. Another important set of methods within the space of contrastive learning use a *pretext task* to contrast similar data-points with randomly selected ones [23] [9]. However, most of the standard data-augmentation schemes [23] used to define the pretext tasks for these approaches have been shown to be not as effective for scene understanding [8].

To address these challenges, we propose a novel contrastive learning approach to find a general-purpose scene representation that is effective for a variety of scene understanding tasks. Our **key intuition** is that commonly available movie metadata (*e.g.*, co-watch, genre, synopsis) can be used to effectively guide the process of learning a generalizable scene representation. Specifically, we use such movie metadata to define a measure of movie-similarity, and use it during contrastive learning to limit our search for positive scene-pairs to only the movies that are considered similar to each other. This allows us to find positive scene-pairs that are not only visually similar but also semantically relevant, and can therefore provide us with a much richer set of geometric and thematic data-augmentations compared to previously employed augmentation schemes [23] [8] (see Figure 1 for illustration). Furthermore, unlike previous contrastive learning approaches that mostly focus on images [23] [9] [13] or shots (§ 3 for definition) [8], our approach builds on the recent developments in vision transformers [12] to allow using variable-length multi-shot inputs. This enables our method to seamlessly incorporate the interplay among multiple shots resulting in a more general-purpose scene representation.

Using a newly collected internal dataset MovieCL30K containing $30,340$ movies to learn our scene representation, we demonstrate the flexibility of our approach to handle both individual shots as well as multi-shot scenes provided as inputs to outperform existing state-of-the-art results on diverse downstream tasks using multiple public benchmark datasets [53] [26] [42]. Furthermore, as an important practical application of long-form video understanding, we apply our scene representation to another newly collected dataset MCD focused on large-scale video moderation with $44,581$ video clips from $18,330$ movies and TV episodes containing sex, violence, and drug-use activities. We show that learning our general-purpose scene representation is crucial

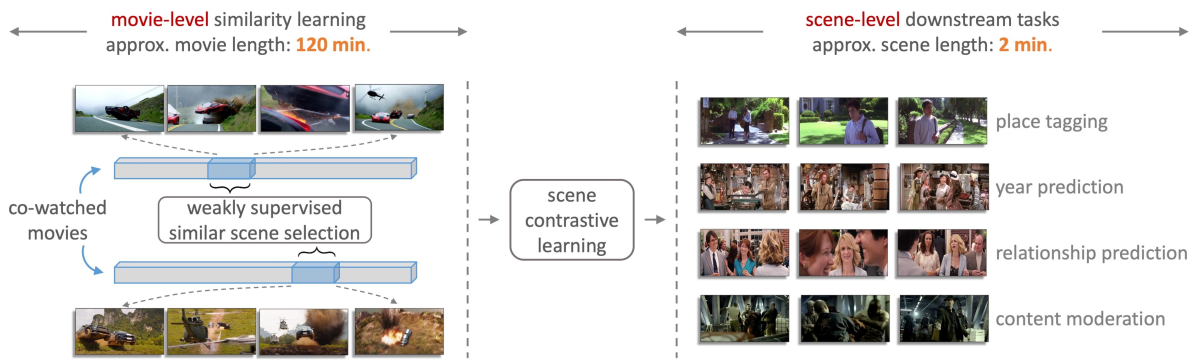

Figure 1. **Approach Overview –** We employ commonly available movie metadata (*e.g.*, co-watch, genre, synopsis) to define movie similarities. The figure illustrates a pair of similar movies where movie similarity is defined based on co-watch information, *i.e.*, viewers who watched one movie often watched the second movie as well. Our approach automatically selects thematically similar scenes from such similar movie-pairs and uses them to learn scene-level representations that can be used for a variety of downstream tasks.

to recognize such age-appropriate video-content where existing representations learned for short-form action recognition or image classification are significantly less effective.

2. Related Work

a. Long-Form Video Understanding: Recent work on semantic understanding of long-form movies and TV episodes have used multi-shot scenes as their processing unit. For example in MovieNet [26], manually annotated multi-shot scenes were used to train various recently proposed models [7] [16] to evaluate their performance on multiple tasks related to scene tagging, *e.g.*, recognizing places or actions in those scenes. In [36], multi-shot clips of movies and TV episodes were categorized into 25 event classes for their temporal localization. The results in [36] show that state-of-the-art event localization models [57] [59] do not perform as well on long-form movies and TV episodes compared to their performance on short-form video datasets like THUMOS14 [29]. A long-form video understanding (LVU) dataset was recently proposed in [53] with nine different tasks related to semantic understanding of video-clips that were cut-out from full-length movies. This work also proposed an object-centric transformer-based video recognition architecture that outperformed SlowFast [16] and VideoBert [46] models on their LVU [53] dataset. To complement existing datasets focusing on regular everyday activity-categories, we collect a new dataset focusing on video moderation of sensitive activities including sex, violence, and drug-use, and show how our scene representation can be applied to recognize these activities.

b. Contrastive Learning: As an important subset of self-supervised learning [30], contrastive learning [32] attempts to learn data representations by contrasting similar data against dissimilar data while using a contrastive loss. Recent approaches for contrastive learning [23] [9] have successfully been extended for a variety of applications in image classification [47] [25] [20] as well as video understanding [22] [8] [61]. More recently, various visio-linguistic models [41] have adopted natural language supervision during contrastive learning to learn correspondences between image and text pairs, which has achieved impressive results especially for zero-shot image classification tasks. Most of these approaches however focus on images or short-videos, and our experiments show that they do not perform as well on tasks related to long-form video understanding.

c. Vision Transformer: Derived from the work on self-attention [49], transformers have been extensively studied in NLP [52] [11]. Building on this line of work, recently proposed vision transformer (ViT) [12] has successfully exceeded state-of-the-art results when pre-trained on large-scale datasets. The data-efficiency of ViT was recently improved in [48] by using token-based distillation for model training. More recently, ViT-based architectures were adopted for video recognition, *e.g.*, TimeSformer [3], ViViT [2], and MViT [14]. These models take images or short-form videos as inputs, and therefore cannot be effectively applied to longer videos without additional tuning.

3. Method

For consistency, let us define a **shot** as a series of frames captured from the same camera over a consecutive period of time [45]. We define a **scene** as a series of consecutive shots without requiring manually annotated scene-boundaries. Note that our definition of scenes is less constrained than what has previously been used [26] [42] [8], and enables our approach to work seamlessly with settings that do [26] or do not have [53] scene boundaries available.

Our approach consists of four key steps. First, we train a shot encoder to provide appearance-based representations using unlabeled movie-shots. Second, we use commonly available movie metadata (*e.g.*, co-watch, genre or synopsis) to define a movie-level similarity $\mathcal{S}$ (see § 4.1.1 for details). We use $\mathcal{S}$ as pseudo-labels to train a movie-level encoder that maximizes the similarity between scenes

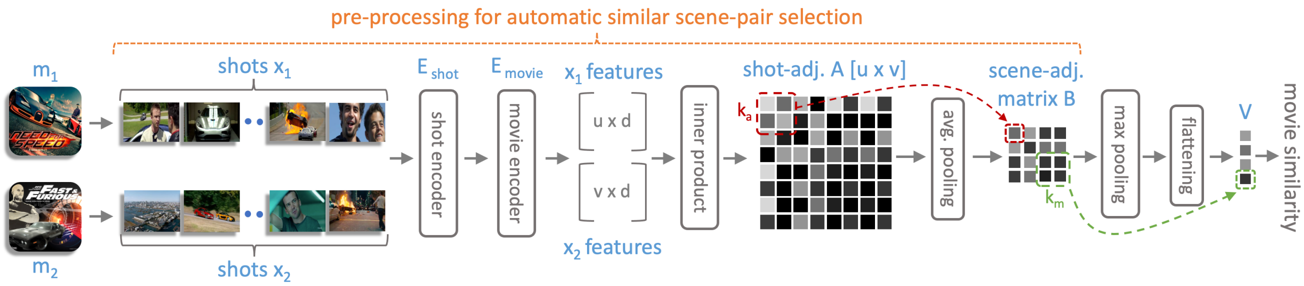

Figure 2. **Movie Similarity Learning –** Shots of a movie-pair $\mathbf{m}_1$ and $\mathbf{m}_2$ are first provided to $\mathbf{E}_{\text{shot}}$ and $\mathbf{E}_{\text{movie}}$ to extract $\mathbf{d}$-dimensional features with numbers of $\mathbf{u}$ and $\mathbf{v}$ shots, respectively. We take the dot-product of shot-feature matrices followed by successive pooling and flattening operations to get an output vector $\mathbf{V}$, which is then regressed against defined movie-level similarity $\mathcal{S}$ between $\mathbf{m}_1$ and $\mathbf{m}_2$.

Figure 3. **Contrastive Scene Representation Learning –** Given a pair of similar movies $\mathbf{m}_1$ and $\mathbf{m}_2$ as determined by $\mathcal{S}$, we use a set of pre-processing operations (as shown in Figure 2) to select similar scene-pairs (see supplementary material for more examples). This process is applied to all similar movie-pairs to get a set of paired scenes that is used for scene-level contrastive learning.

from similar movies (as defined by $\mathcal{S}$). Third, we use this trained movie-level encoder to select similar scene-pairs for contrastive learning of scene encoder. Lastly, we use the learned scene representation for downstream tasks.

3.1. Movie Similarity Learning

3.1.1 Shot Encoder

As it is generally inefficient to directly train a network with all shots from movies as input in an end-to-end manner [42] [26] [8], we divide movie-level training into two stages, *i.e.*,: (a) shot encoder training, and (b) movie encoder training. During the first stage, we train a shot encoder end-to-end to provide features mostly focusing on appearance-related information without much contextual meaning.

Given a query shot $\mathbf{x_t}$ at time t, we use its neighboring shots $\mathbf{x_{t-2}}$ and $\mathbf{x_{t+2}}$ as a pair of potential positive keys. We train our shot encoder to distinguish a positive key shot from randomly selected negative key shots, where the positive logits are defined as:

$$\max(\mathbf{E_{shot}}(\mathbf{x_t}) \cdot \mathbf{E_{shot}}(\mathbf{x_{t\text{-}2}}), \mathbf{E_{shot}}(\mathbf{x_t}) \cdot \mathbf{E_{shot}}(\mathbf{x_{t\text{+}2}})) \quad (1)$$

3.1.2 Movie Encoder

After training the shot encoder, we fix it to extract shot representations and add a light-weight encoder on top it as movie encoder. As illustrated in Figure 2, we divide a given movie-pair $\mathbf{m}_1$ and $\mathbf{m}_2$ into their constituent shots [44] $\mathbf{x}_1$ and $\mathbf{x}_2$ respectively. We extract features of these shots using the fixed shot encoder $\mathbf{E}_{\text{shot}}$, so that $\mathbf{x}_1$ and $\mathbf{x}_2$ are represented by two feature matrices $\mathbf{E}_{\text{shot}}(\mathbf{x}_1)$ and $\mathbf{E}_{\text{shot}}(\mathbf{x}_2)$ respectively, which are passed through an encoder with learnable parameters $\mathbf{E}_{\text{movie}}$, followed by their dot-product to create a shot-adjacency matrix $\mathbf{A}_{\mathbf{x}_1,\mathbf{x}_2}$, *i.e.*,

$$\mathbf{A}_{\mathbf{x}_1,\mathbf{x}_2} = \mathbf{E}_{\text{movie}}(\mathbf{E}_{\text{shot}}(\mathbf{x}_1)) \cdot \mathbf{E}_{\text{movie}}(\mathbf{E}_{\text{shot}}(\mathbf{x}_2)) \quad (2)$$

Note that entries in $\mathbf{A}_{\mathbf{x}_1,\mathbf{x}_2}$ represent pair-wise similarities between shots in movies $\mathbf{m}_1$ and $\mathbf{m}_2$.

Using $\mathbf{k}_\text{a}$ contiguous shots as scenes, we convert the shot-adjacency matrix $\mathbf{A}_{\mathbf{x}_1,\mathbf{x}_2}$ into a scene-adjacency matrix $\mathbf{B}_{\mathbf{x}_1,\mathbf{x}_2}$ by applying average pooling with kernel size $\mathbf{k}_\text{a}$ and stride $\mathbf{s}_\text{a}$ on $\mathbf{A}_{\mathbf{x}_1\mathbf{x}_2}$ to calculate the average values in each $\mathbf{k}_\text{a} \times \mathbf{k}_\text{a}$ window in $\mathbf{A}_{\mathbf{x}_1,\mathbf{x}_2}$. Furthermore, we apply max pooling with kernel size $\mathbf{k}_\text{m}$ and stride $\mathbf{s}_\text{m}$ on $\mathbf{B}_{\mathbf{x}_1,\mathbf{x}_2}$ followed by flattening the output to get the vector $\mathbf{V}_{\mathbf{x}_1,\mathbf{x}_2}$. Intuitively, this step finds the most similar scene-pairs between $\mathbf{x}_1$ and $\mathbf{x}_2$, where each value in $\mathbf{V}_{\mathbf{x}_1,\mathbf{x}_2}$ represents the highest similarity of the scene-pair in a neighborhood of $\mathbf{k}_\text{m} \times \mathbf{k}_\text{m}$ scenes in $\mathbf{B}_{\mathbf{x}_1,\mathbf{x}_2}$. Finally, we learn the projection between $\mathbf{V}_{\mathbf{x}_1,\mathbf{x}_2}$ and the output using layer $\mathbf{L}_{\text{output}}$, where our goal is to predict if $\mathbf{x}_1$ and $\mathbf{x}_2$ are similar at movie-level. That is, if $\mathbf{x}_1$ and $\mathbf{x}_2$ are shots from similar movies as defined by $\mathcal{S}$, our target label is 1, and 0 otherwise. During training, we use cross-entropy loss to update $\mathbf{E}_{\text{movie}}$ and $\mathbf{L}_{\text{output}}$ while $\mathbf{E}_{\text{shot}}$ remains unchanged.

A useful way to interpret our movie similarity learning step is to view it from a multiple instance learning (MIL) perspective [5]. The shot-adjacency matrix $\mathbf{A}_{\mathbf{x}_1,\mathbf{x}_2}$ can be considered as a *bag*, while similar scene-pairs can be thought of as positive instances. Unlike some recent MIL-based approaches [1] [17] [34], we simplify our learning process by adopting standard network operations and loss function instead of specially designed ones, which allows us to use existing well-engineered implementations to get better efficiency and scalability in practice.

3.2. Scene Contrastive Learning

Given a similar movie-pair as defined by $\mathcal{S}$, we apply $\mathbf{E}_{\text{movie}}(\mathbf{E}_{\text{shot}}(\cdot))$ to their constituent shots, followed by successive pooling operations to find the scene adjacency matrix $\mathbf{B}$, which is used to rank the similarity of scene-pairs. We apply this selection process to all pairs of similar movies to get a collection of paired scenes $\mathbf{P}_{\text{scene}}$ that is used for scene-level contrastive learning. Our contrastive learning approach is shown in Figure 3, and explained below.

3.2.1 Scene Encoder

As the inputs to our encoder $\mathbf{E}_{\text{scene}}$ for scene contrastive learning are multi-shot sequences, it is important to design $\mathbf{E}_{\text{scene}}$ so that it can effectively model the various relationships among input shots. To this end, we build on recent work of ViT [12], and propose a transformer based $\mathbf{E}_{\text{scene}}$ that treats patches in input shots as tokens.

Specifically, following [12] [41], for a k-frame shot of dimension (k, w, h, c), we first divide it into a sequence of $(k, \frac{w}{p}, \frac{h}{p}, c)$ patches, where (p, p) is the size of each patch. To input the shot to a transformer with latent vector size of D dimensions, we apply $\mathrm{D} \times c$ convolutional kernels with kernel size (p, p) and stride (p, p) to the $(k, \frac{w}{p}, \frac{h}{p}, c)$ patches. This converts the shot into patch embeddings with dimension $(k, \frac{h}{p}, \frac{w}{p}, \mathrm{D})$, which is further flattened to a $(k, \mathrm{N}, \mathrm{D})$-dimensional tensor where $\mathrm{N} = (w \cdot h)/p^2$. Furthermore, we prepend a learnable embedding to the patch embeddings similar to the class token used in [11]. After permutation, the result is $(\mathrm{N}{+}1, \mathrm{D})$-dimensional patch embeddings for each of the k frames. We add $(\mathrm{N}{+}1, \mathrm{D})$-dimensional positional embeddings to patch embeddings to retain positional information, and pass them to successive multi-headed self-attention (MSA) layers as done in [12].

Similarly, for the input of a scene with n shots, we first divide it into a sequence of $(n \cdot k, \frac{w}{p}, \frac{h}{p}, c)$ patches. After convolution and flattening, we get $(\mathrm{N} \cdot n \cdot k, \mathrm{D})$-dimensional patch embeddings. Notice that this is different from the dimension of a frame, and does not match with the dimension of positional embeddings. Inspired by [12] where they mentioned that positional embeddings can be interpolated to take input of higher resolutions, we propose to generalize this property from frame to shot and scene. That is, we interpolate the N-dimensional positional embeddings to $\mathrm{N} \cdot n \cdot k$, excluding the 1 dimension corresponding to class token, and add the interpolated positional embeddings to patch embeddings before providing them to MSA layers.

This operation offers flexibility and efficiency for downstream tasks. That is, using 2-D interpolation to the positional embeddings of our encoder allows it to take variable-length shot-sequences as inputs. This enables our approach to be applicable to the common setting where the lengths of multi-shot sequences in contrastive learning are different from those available in downstream tasks. Moreover, as using 2-D interpolation to the positional embeddings of our encoder does not add extra computation along the temporal dimension, it allows us to perform training efficiently which is particularly important for large-scale settings.

3.2.2 Contrastive Learning

Our scene contrastive learning step follows some of the recent works [23] [8] with two key differences: (a) unlike the image or shot-augmentation focused pretext tasks previously used, we define scene-level pretext task that uses commonly available movie metadata making it more effective for long-form video understanding, and (b) our use of ViT-based [12] scene encoder allows variable-length inputs and the possibility to adopt large-scale pre-trained models, which is not something that previously used ResNet-based encoders [8] could offer.

Specifically, we define our pretext task as a dictionary look-up for the scenes selected at the end of our movie similarity learning step. That is, given a query scene q, its positive key scene k_0 is determined in $\mathbf{P}_{\text{scene}}$, and the objective is to find k_0 among a set of random scenes $\{k_1, k_2, ..., k_K\}$. The problem is converted to a $(K+1)$-way classification task by calculating similarity with dot-product, and we use InfoNCE as the contrastive loss function [40]:

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

where $f(\cdot|\theta_q)$ is the query encoder with parameters θ_q updated during back-propagation, $g(\cdot|\theta_k)$ is the key encoder with parameters θ_k learned by momentum update as done in [23], and τ is the temperature as illustrated in [55].

During contrastive learning, the scene encoder $\mathbf{E}_{\text{scene}}$ is the query encoder q without the last fully-connected layer, and is updated based on the selected similar scenes $\mathbf{P}_{\text{scene}}$. After training converges, $\mathbf{E}_{\text{scene}}$ is the outcome from this stage, which can then be used for downstream tasks.

4. Experiments

We first discuss one of our newly collected movie datasets MovieCL30K that we used to learn our scene representation followed by describing the implementation details of our algorithm. We then present comparative results on multiple benchmark datasets [53] [26] [42] demonstrating significant gains of our approach over existing state-of-the-art models for a diverse set of downstream tasks in LVU [53] and MovieNet [26]. To further demonstrate the generalizability of our learned scene representation, we test it on another newly collected dataset MCD focused on large-scale moderation of age-sensitive activities and show substantial gains over existing state-of-the-art approaches.

4.1. Dataset and Implementation

4.1.1 MovieCL30K

We compiled a new dataset called MovieCL30K with $30,340$ movies from 11 genres with the genre-distribution of [action, drama, comedy, horror, doc, foreign, animation, kids, sci-fi, romance, other] being [27.4%, 21.6%, 17.5%, 6.5%, 4.6%, 4.6%, 4.1%, 3.1%, 2.8%, 1.2%, 6.1%]. We empirically compare commonly available movie metadata including: (i) co-watch, (ii) genre, and (iii) synopsis.

We collect movie co-watch information based on the watch history of movies. That is, for each movie $\mathbf{m}_1$, we keep records of movies that are watched after $\mathbf{m}_1$, and then rank the records to build a set of movies that are most frequently watched after $\mathbf{m}_1$. To use genre to find movies similar to an input movie, we randomly select movies with the same genre as the input movie. Lastly, to use synopsis for movie-level similarity, we first extract textual-embeddings of synopsis using a pre-trained natural language processing model [37], and then compute pair-wise movie-similarities as the dot-product of their textual-embeddings followed by selecting the closest movies. When using any of the aforementioned movie-similarity measures, we select three most similar movies on average for each movie in MovieCL30K.

4.1.2 Implementation Details

We divide movies in MovieCL30K into their constituent $\sim$33 million shots, and only keep one frame per shot for efficiency. We first use ViT [12] as our shot encoder $\mathbf{E}_{\text{shot}}$ and train it with 30% of all shots forming query and key pairs. Then we fix and use $\mathbf{E}_{\text{shot}}$ to extract representations of all shots as shown in Figure 2. The movie-level encoder $\mathbf{E}_{\text{movie}}$ comprises of two fully-connected layers each with 512 dimensions, and a dropout layer with 0.5 probability between them. We keep length of each movie to be $1,024$ shots, and zero-pad the movies shorter than that. Kernel size for both max and average pooling are set to 16, with a stride of 8.

After movie-level similarity learning (Figure 2), we only keep the top 50% of most similar scene-pairs selected in each movie-pair because we empirically found that the scene-pairs can get noisier and become less informative beyond this ratio. We follow the implementation in [23] for momentum contrastive learning and use ViT [12] as our scene encoder instead of ResNet [24] used in [23]. Moreover, we optimize with AdamW [38] instead of SGD [19]. For scene representation learning, we interpolate the positional embeddings (see § 3.2.1) so that the encoder can take variable-length multi-shot sequences as input.

For supervised learning on downstream tasks, unless otherwise specified, we use a simple multilayer perceptron (MLP) with two hidden layers, where the output layer is modified for each various classification or regression task.

4.2. Comparisons on LVU Benchmark

To ensure that there is no overlap between pre-training and downstream evaluation, during the representation learning step, we exclude movies from MovieCL30K that have the same IMDb IDs as the ones in validation and test sets of compared benchmark datasets.

We first evaluate our model using the benchmark LVU dataset [53] which contains nine diverse tasks including: place (scene), director, relationship, way-of-speaking, writer, year, genre, view-count, and like-ratio. The total number of labeled videos in LVU dataset [53] is $\sim$11K with the splits of training, validation, and testing sets to be 70%, 15%, and 15%, respectively. Following [53], view-count and like-ratio are evaluated by mean-squared-error in regression (lower is better), while other seven tasks are evaluated using top-1 accuracy in classification (higher is better).

4.2.1 Ablation Study

To assess the effectiveness of different components in our model, we present the ablative comparisons in Table 1 regarding factors including: 1) number of movies used during pre-training, 2) whether the shot encoder is initialized using CLIP [41] visual encoder and kept fixed or randomly initialized and trained by our method, 3) whether the scene encoder is initialized with CLIP [41] weights or randomly, 4) influence of number of shots in each scene and number of frames in each shot, 5) different backbone architectures, 6) hyper-parameters including batch size and input resolution, 7) how long it takes for the training to complete compared to row 1 as one unit time, 8) the metadata used to define similar movie-pairs. Results on validation-set of nine tasks in LVU [53] are presented for each setting. The insights from these comparisons are summarized as follows, where the results are compared against row 1 as the base case.

a. Training scale is critical for model performance. From rows 1 to 5 in Table 1, we compare different scales of pre-training. We can see that our model can work with as little as 3K pre-training movies, but it takes about 15K movies for our method to compare favorably with previous state-of-the-art results [53]. Since we do not need any manually labeled data for pre-training, we can scale our approach for better performance relatively more easily.

b. Number of shots is important. From rows 6 and 7, comparing with row 1, the performance would be improved when we increase the number of shots per scene. This is reasonable since we can incorporate more information with more shots. However, when we further increase the number to 16 shots per scene in row 8, the performance drops. This is possibly because with the longer context, some scenes may cross scene boundaries, and thus provide noisy or irrelevant information from different scenes. Also,

	pre-train movies	shot enc.	scene init.	# of shots -frames	backbone	hyper-params bat.-res.	unit time	metadata	classification ↑ place	director	relation	speak	writer	year	genre	regression ↓ view	like
Various pre-training scales																	
1	30K	random	random	9-1	ViT-B/16	128-128	1	co-watch	63.2	69.8	68.7	40.1	60.1	54.1	55.5	2.62	0.161
2	15K	random	random	9-1	ViT-B/16	128-128	0.5	co-watch	59.1	66.3	63.3	37.5	55.9	50.3	52.7	3.02	0.254
3	7.5K	random	random	9-1	ViT-B/16	128-128	0.25	co-watch	55.7	60.9	59.4	35.8	52.7	48.3	48.8	3.19	0.307
4	3K	random	random	9-1	ViT-B/16	128-128	0.1	co-watch	45.3	49.5	52.2	34.6	47.6	41.2	43.6	3.47	0.352
5	300	random	random	9-1	ViT-B/16	128-128	0.01	co-watch	31.2	29.8	35.7	27.5	30.1	28.9	34.3	4.55	0.496
Different numbers of shots and frames																	
6	30K	random	random	1-1	ViT-B/16	128-128	1	co-watch	46.5	48.2	44.1	30.3	31.1	34.4	45.8	3.89	0.401
7	30K	random	random	4-1	ViT-B/16	128-128	1	co-watch	55.6	60.1	57.2	33.4	50.3	49.8	51.1	3.07	0.246
8	30K	random	random	16-1	ViT-B/16	128-128	1	co-watch	62.4	66.2	65.3	39.7	56.2	54.3	52.4	2.93	0.173
9	30K	random	random	9-3	ViT-B/16	128-128	3	co-watch	63.5	70.1	69.3	40.4	59.7	54.9	54.7	2.59	0.159
Different similarity measures																	
10	30K	random	random	9-1	ViT-B/16	128-128	1	random	49.2	48.8	51.5	34.6	37.2	36.7	49.1	3.94	0.454
11	30K	random	random	9-1	ViT-B/16	128-128	1	genre	62.9	58.1	66.3	39.5	47.3	45.2	56.2	3.85	0.351
12	30K	random	random	9-1	ViT-B/16	128-128	1	synopsis	56.7	54.1	65.3	36.8	45.9	34.5	50.7	3.87	0.471
13	30K	random	random	9-1	ViT-B/16	128-128	1	gen.+syn.	64.1	65.5	69.6	41.5	52.3	51.1	56.7	2.94	0.199
Comparisons on weight initialization and backbone																	
14	30K	CLIP	CLIP	9-1	ViT-B/16	128-128	1	co-watch	61.2	65.3	67.4	43.7	54.1	53.5	56.3	2.75	0.178
15	3K	random	random	9-1	MViTv2-S	32-224	1	co-watch	47.6	51.2	51.3	35.3	45.1	44.7	43.7	3.51	0.343

Table 1. Ablation study results on validation-set of benchmark LVU dataset [53]. Note that for classification tasks higher is better, and for regression tasks lower is better.

	models	pre-train data	modalities	frames /scene	params sup. learn	classification ↑ place	director	relation	speak	writer	year	genre	regression ↓ view	like
End-to-end learning approaches														
1	Video Bert [46]	30K LVU movie clips	visual	60	∼8.77M	54.9	47.3	52.8	37.9	38.5	36.1	51.9	4.46	0.320
2	SlowFast R101 [16]	30K LVU movie clips	visual	60	∼44M	54.7	44.9	52.4	35.8	36.3	52.5	53	3.77	0.386
3	OT [53]	30K LVU movie clips	visual	60	∼27M	56.9	51.2	53.1	39.4	34.5	39.1	54.6	3.55	0.230
4	ViS4mer [27]	30K LVU movie clips	visual	60	∼3.6M	67.4	62.6	57.1	40.7	48.8	44.7	54.7	3.63	0.260
5	Hierarchical OT [56]	240K Kinetics +23.5K VidSitu	visual	16	∼27M	44.1	40.1	50.9	34.1	31.4	29.6	51.1	4.88	0.353
Representation-based approaches														
6	CLIP [41]	400M image-text pairs	vis.+text	9	∼0.7M	52.9	56.2	56.1	36.7	37.8	46.4	50.9	3.85	0.411
7	Merlot Reserve [58]	20M Youtube videos	vis.+text+aud.	8	∼0.7M	59.2	54.4	60.0	41.1	38.5	49.7	54.6	3.03	0.217
8	ShotCoL [8]	2.5M movie shot pairs	visual	9	∼1.3M	45.3	49.5	46.7	31.1	29.8	36.7	43.1	4.79	0.397
9	Bridge Former [18]	3.3M image+2.5M video	vis.+text	4	∼0.4M	62.8	55.7	60.4	40.9	49.7	41.4	52.9	3.97	0.312
10	Ours	2.5M movie scene-pairs	visual	9	∼0.7M	**68.2**	**70.9**	**71.2**	**42.2**	**53.7**	**57.8**	**55.9**	**2.79**	**0.192**
						+0.8	**+8.3**	**+10.8**	**+1.1**	**+4.0**	**+5.3**	**+1.2**	**-0.24**	**-0.025**

Table 2. Quantitative comparisons on test-set of benchmark LVU dataset [53]. Our approach is evaluated on nine diverse tasks and compared against multiple state-of-the-art models. Note that for classification tasks higher is better, and for regression tasks lower is better.

the number of scene-pairs would decrease with more shots per scene, and thus provide fewer training samples for scene contrastive learning, which may further reduce the effectiveness of the learned scene representations. Adding more frames per shot as in row 9 can also increase the performance. However, when we have more frames to process, it takes significantly more time for training with only marginal performance gain. Thus, we choose to use 9 shots per scene with 1 frame per shot for all future experiments as default.

c. Movie metadata matters. In rows 10 to 12, we compare the representations learned using different type of movie metadata. We can see that co-watch, genre and synopsis are all effective measures for movie similarity. Using co-watch can achieve overall better results, which is likely due to the fact that it can incorporate more complex relationships between movies, and thus provide more diverse scene-pairs for representation learning. Moreover, when we concatenate the features separately learned by genre and synopsis, it offers similar or even better accuracy compared to co-watch, which indicates that further improvements can be achieved when combining scene representations learned from different metadata information. However, as shown in row 10, randomly picking movie-pairs from the 30K movies and applying our approach to them does not offer results that are comparable to what we get when using other types of metadata. This indicates that informative and meaningful metadata is necessary for our model to perform well.

d. Our method does not rely on pre-trained weights. In row 14, we swap our shot encoder to the visual encoder in CLIP [41] and use CLIP weights as initialization for scene encoder. We can see that the performance is comparable to random initialization, showing that our whole framework can be trained from scratch using solely movie metadata.

e. Model-complexity and training-scale trade-off. When comparing with different backbone architecture MViTv2-S [14] [35] in row 15, we notice that while recent architectures can improve performance, it may take significantly more time for training. For example, even when the number of parameters in ViT-B/16 (87.2M) is more than the one in MViTv2-S (26.1M) [14] [35], the expected time to complete the training of MViTv2-S with their default setting is more than $10\times$ that of ViT-B/16 due to smaller batch size and larger input size. To keep $\sim$1 unit time for training of MViTv2-S, we can only use 3K pre-training data. It performs better than row 4 using same amount of data but only requires 0.1 unit time, and it is not comparable with row 1 that can use $10\times$ of data with 1 unit time. Thus, we keep our setting on row 1 as default.

4.2.2 Comparing with State-of-the-art Results

In Table 2 we compare our approach with state-of-the-art results on LVU [53] test-set. There are two main parts of Table 2 where each model is considered as either an end-to-end learning or representation-based approach. For end-to-end methods, they generally use a smaller scale of pre-training data, followed by end-to-end fine-tuning on downstream tasks in LVU. Thus, they require to train a higher number of parameters during supervised training. A higher number of frames per scene is also used in end-to-end models, which enables them to incorporate more information.

For representation-based approaches, we compare with some recent large-scale pre-training models using various scales and modalities of data. We extract features from their fixed visual encoders and learn a simple MLP for downstream tasks, which significantly reduces the number of parameters needed during supervised learning. We can see that our method is significantly different from other existing representation-based approaches in terms of required modalities. That is, most existing methods rely on cross-modality information to learn representations, and this requires a more complex procedure for data process and collection. The only approach using single modality is ShotCoL [8]. However, since they learned shot-level representations that are most effective for scene boundary detection tasks, when applied to semantic understanding of content with longer context, they are not performing as well.

Overall, we can see that comparing with both end-to-end and representation-based approaches, our model can comfortably outperform other state-of-the-art results across different tasks, with a large margin on many of them. These comparisons further validate the effectiveness of our learned scene representations.

Models		mAP
Fine-tuning	TSN [50]	8.33
	I3D [7]	7.66
Feature +MLP	ImgNet [10]	7.04
	Place [60]	8.76
	CLIP [41]	9.26
	Ours	**12.17**

Table 3. Comparisons with state-of-the-art results on MovieNet place scene tagging using methods based on: (a) action recognition models fine-tuned for this task, and (b) commonly used pre-trained representations for generic downstream tasks.

Models	Pre-train data	AP
SimCLR [9]	same-domain	41.65
MoCo [23]	same-domain	42.51
ShotCoL [8]	same-domain	53.37
SCRL [54]	same-domain	54.82
BaSSL [39]	same-domain	**57.40**
LGSS [42]	cross-domain	47.1
CLIP [41]	cross-domain	47.4
ShotCoL [8]	cross-domain	48.4
Ours	cross-domain	**55.03**

Table 4. Comparisons with state-of-the-art results on SBD. Our approach outperforms all methods trained in cross-domain and provides competitive results to models trained in same-domain.

4.3. MovieNet Benchmark

MovieNet is a large-scale movie understanding dataset with 1,100 movies covering a variety of tasks [26]. However, unlike LVU [53] where videos are publicly available, the movies in MovieNet are not publicly released yet due to copyright issues. Only keyframes from each shot are provided, which makes tasks that demand high frame-rate (*e.g.*, action recognition) infeasible. We therefore focus our comparisons to two scene understanding related tasks in MovieNet that do not demand particularly high frame-rate, *i.e.*: (i) place tagging, and (ii) scene boundary detection.

a. Place Tagging: There are 90 categories of places with 19.6K place tags in MovieNet [26]. The place tagging problem is formulated as a multi-label classification task where each manually labeled scene can have multiple place tags. The results are evaluated by mean average precision (mAP) on the test-set as shown in Table 3. We compare our approach with two sets of models: (i) fine-tuning models, and (ii) feature-based approaches. The results on fine-tuning are reported by [26] where they mentioned that standard action recognition models were adopted. For feature-based approaches, we compare our encoder with CLIP visual encoder [41] and ResNet50 [24] pre-trained on the Place dataset [60]. We can see that our model can significantly outperform all other compared methods on this task.

b. Scene Boundary Detection: Scene boundary detection (SBD) is a challenging and important problem for semantic movie understanding [42] [8]. We show that our learned general-purpose scene encoder can be seamlessly applied to this task and provide competitive results to state-of-the-art models. Following [8], we use the subset of 318 movies from MovieNet [42] [26] where manually annotated scene boundaries are provided. The training, testing and validation splits are also provided with 190, 64 and 64 movies respectively. The problem is formulated as binary classification to predict if a shot boundary is also a scene boundary,

Models	Pre-training data	sex	violence	drug-use	average
SlowFast R50 [16]	K400 [31]	63.9	46.5	49.4	53.2
SlowFast R101 [16]	K600 [6] +AVA2.2 [33]	61.0	57.3	54.0	57.4
X3D-L [15]	K400 [31]	69.2	49.4	56.4	58.3
CLIP [41]	image-text pairs	78.5	62.1	55.1	65.2
Ours	MovieCL30K	**81.5**	**70.2**	**61.8**	**71.1**

Table 5. Comparison of our approach with state-of-the-art pre-trained models on MCD dataset.

and the results are evaluated by average precision (AP) [26] [8]. Following ShotCoL [8], we use a four-shot sequence as a sample where representations for each shot are independently extracted and concatenated. Notice that in addition to Place Tagging and LVU tasks where our encoder can take multi-shot clips as input, for SBD, our encoder can take individual shots as input which demonstrates the flexibility of our approach for different types of inputs.

We report the AP results of the different compared approaches in Table 4, where approaches are categorized depending on whether the pre-training data comes from the same domain as the downstream task. For example, there are four encoders in LGSS [42] pre-trained in supervised manner on datasets that are different from MovieNet [26], and thus it is the cross-domain setting. For SCRL [54], BaSSL [39] and MoCo [23] implemented in [8], the representations were learned on the full MovieNet [26] dataset with $1,100$ movies, and then applied to the subset of 318 movies on SBD, making it a same-domain setting. ShotCol [8] provided results for both settings with a significant performance gap between them.

As shown in Table 4, our approach outperforms state-of-the-art models pre-trained in cross-domain, and provides competitive results to models [8] [54] [39] that are dedicatedly designed for SBD and pre-trained in same-domain.

4.4. Video Moderation

Large-scale video moderation is one of the most pressing challenges faced by video streaming services. However, most existing action recognition datasets (*e.g.*, Kinetics [6, 31], THUMOS14 [29]) do not cover movies and TV episodes well, while existing movie datasets (AVA [21], MovieNet [26]) are not including age-sensitive activities. To this end, we focus on video moderation particularly to detect sex, violence, and drug-use activities (see supplementary materials for examples) in movies and TV episodes.

a. Mature Content Dataset (MCD)**:** We constructed the MCD dataset by using Amazon SageMaker Ground Truth as our labeling tool. We provided annotation guidelines to annotators consisting of 17 instructions for sex, 34 for violence and 14 for drug-use. Examples of these instructions include: “intentional murder and/or suicide that involves bloody injury” for violence, and “strip-tease or erotic dancing with full nudity” for sex.

All our annotators were fluent English speakers which enabled them to understand their provided instructions. All annotators went through multiple training sessions to prepare for their assignment. Each annotator was asked to label sample-videos and received feedback on their labeling quality before they could start to label independently. Additionally, 20% of the labeled samples were randomly selected and taken through a mandatory review to ensure quality.

MCD contains $44,581$ clips from $18,330$ movies and TV episodes with $4,580$, $8,248$, $8,271$, and $23,482$ clips containing sex, violence, drug-use, and none-of-the-above activities respectively. These clips have a mean duration of 5 seconds, and were divided into training, validation, and testing sets with 70%, 10%, and 20% ratio respectively.

b. Results: We compare our representation with the ones from state-of-the-art models [16] [14] [15] [41] pre-trained on action recognition [31] [6] [21] as well as image classification datasets [41]. Specifically, we extract embeddings of MCD using different pre-trained encoders and use them as input to train a 4-class MLP for predicting age-appropriate activities. Table 5 presents the AP results on our test data, and shows that our scene representation outperforms the alternatives by a large margin. Notice that CLIP visual feature performed close to our representation on the task of sex but not other tasks. This might indicate that in the pre-training data of CLIP, there might be a considerable amount of image-text pairs related to sex concepts. Similar observations [4] have been made on other large-scale datasets [43].

5. Conclusions and Future Work

We presented a novel contrastive learning approach that uses movie metadata to learn a general-purpose scene-level representation. Our approach adopts recent developments in vision transformer, where by incorporating the interpolation of positional embedding, we were able to train a scene encoder that can take variable-length multi-shot inputs. We empirically demonstrated the effectiveness of our approach using different types of movie metadata including co-watch, genre, synopsis. Results on a variety of tasks from multiple benchmark datasets highlight the strengths of our approach. We also presented a new application of our scene representation for video moderation focused on three age-appropriate activities, where our approach outperformed state-of-the-art action recognition features.

Going forward, we will further improve the efficiency of our approach to help it scale even more. We will also explore other types of movie metadata to find a representation effective for additional downstream tasks. Furthermore, there are additional long-form content (*e.g*, music, news articles, books) where acquiring fine-grained labels is costly. We will apply our approach on these long-form content to enable downstream tasks from multiple domains.

References

[1] Baptiste Angles, Yuhe Jin, Simon Kornblith, Andrea Tagliasacchi, and Kwang Moo Yi. Mist: Multiple instance spatial transformer. In *CVPR*, 2021. 3

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *arXiv:2103.15691*, 2021. 2

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv:2102.05095*, 2021. 2

[4] Abeba Birhane, Vinay Uday Prabhu, and Emmanuel Kahembwe. Multimodal datasets: misogyny, pornography, and malignant stereotypes. *arXiv:2110.01963*, 2021. 8

[5] Marc-André Carbonneau, Veronika Cheplygina, Eric Granger, and Ghyslain Gagnon. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 2018. 3

[6] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv:1808.01340*, 2018. 8

[7] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 2, 7

[8] Shixing Chen, Xiaohan Nie, David Fan, Dongqing Zhang, Vimal Bhat, and Raffay Hamid. Shot contrastive self-supervised learning for scene boundary detection. In *CVPR*, 2021. 1, 2, 3, 4, 6, 7, 8

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1, 2, 7

[10] 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. 7

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*, 2018. 2, 4

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 1, 2, 4, 5

[13] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *ICCV*, 2021. 1

[14] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *ICCV*, 2021. 2, 7, 8

[15] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, 2020. 8

[16] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 2, 6, 8

[17] Jia-Chang Feng, Fa-Ting Hong, and Wei-Shi Zheng. Mist: Multiple instance self-training framework for video anomaly detection. In *CVPR*, 2021. 3

[18] Yuying Ge, Yixiao Ge, Xihui Liu, Dian Li, Ying Shan, Xiaohu Qie, and Ping Luo. Bridgeformer: Bridging video-text retrieval with multiple choice questions. In *CVPR*, 2022. 6

[19] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv:1706.02677*, 2017. 5

[20] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv:2006.07733*, 2020. 2

[21] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018. 8

[22] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. In *ECCV*, 2020. 2

[23] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1, 2, 4, 5, 7, 8

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5, 7

[25] Olivier J Henaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. In *ICML*, 2020. 2

[26] Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *ECCV*, 2020. 1, 2, 3, 4, 7, 8

[27] Md Mohaiminul Islam and Gedas Bertasius. Long movie clip classification with state-space video models. In *ECCV*, 2022. 6

[28] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 2021. 1

[29] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. `http://crcv.ucf.edu/THUMOS14/`, 2014. 2, 8

[30] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *TPAMI*, 2020. 2

[31] 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:1705.06950*, 2017. 8

[32] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. Contrastive representation learning: A framework and review. *IEEE Access*, 2020. 2

[33] Ang Li, Meghana Thotakuri, David A Ross, João Carreira, Alexander Vostrikov, and Andrew Zisserman. The ava-kinetics localized human actions video dataset. *arXiv:2005.00214*, 2020. 8

[34] Bin Li, Yin Li, and Kevin W. Eliceiri. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *CVPR*, 2021. 3

[35] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *CVPR*, 2022. 7

[36] Xiaolong Liu, Yao Hu, Song Bai, Fei Ding, Xiang Bai, and Philip H. S. Torr. Multi-shot temporal event localization: A benchmark. In *CVPR*, 2021. 2

[37] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv:1907.11692*, 2019. 5

[38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 5

[39] Jonghwan Mun, Minchul Shin, Gunsoo Han, Sangho Lee, Seongsu Ha, Joonseok Lee, and Eun-Sol Kim. Bassl: Boundary-aware self-supervised learning for video scene segmentation. In *ACCV*, 2022. 7, 8

[40] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv:1807.03748*, 2018. 4

[41] 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. *arXiv:2103.00020*, 2021. 1, 2, 4, 5, 6, 7, 8

[42] 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 *CVPR*, 2020. 1, 2, 3, 4, 7, 8

[43] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv:2111.02114*, 2021. 8

[44] 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. 3

[45] Robert Sklar. Film: An international history of the medium. *Thames and Hudson*, 1990. 2

[46] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, 2019. 2, 6

[47] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *arXiv:2005.10243*, 2020. 2

[48] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 2

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[50] 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. 7

[51] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys*, 2020. 1

[52] Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. Transformers: State-of-the-art natural language processing. In *Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020. 2

[53] Chao-Yuan Wu and Philipp Krahenbuhl. Towards long-form video understanding. In *CVPR*, 2021. 1, 2, 4, 5, 6, 7

[54] Haoqian Wu, Keyu Chen, Yanan Luo, Ruizhi Qiao, Bo Ren, Haozhe Liu, Weicheng Xie, and Linlin Shen. Scene consistency representation learning for video scene segmentation. In *CVPR*, 2022. 7, 8

[55] Zhirong Wu, Yuanjun Xiong, X Yu Stella, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 4

[56] Fanyi Xiao, Kaustav Kundu, Joseph Tighe, and Davide Modolo. Hierarchical self-supervised representation learning for movie understanding. In *CVPR*, 2022. 6

[57] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *CVPR*, 2020. 2

[58] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. Merlot reserve: Multimodal neural script knowledge through vision and language and sound. In *CVPR*, 2022. 6

[59] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *ICCV*, 2019. 2

[60] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *TPAMI*, 2017. 7

[61] Mohammadreza Zolfaghari, Yi Zhu, Peter Gehler, and Thomas Brox. Crossclr: Cross-modal contrastive learning for multi-modal video representations. In *ICCV*, 2021. 2