Long-range Multimodal Pretraining for Movie Understanding

Dawit Mureja Argaw
KAIST

Joon-Young Lee
Adobe

Markus Woodson
Adobe

In So Kweon
KAIST

Fabian Caba Heilbron
Adobe

Abstract

Learning computer vision models from (and for) movies has a long-standing history. While great progress has been attained, there is still a need for a pretrained multimodal model that can perform well in the ever-growing set of movie understanding tasks the community has been establishing. In this work, we introduce Long-range Multimodal Pretraining, a strategy, and a model that leverages movie data to train transferable multimodal and cross-modal encoders. Our key idea is to learn from all modalities in a movie by observing and extracting relationships over a long-range. After pretraining, we run ablation studies on the LVU benchmark and validate our modeling choices and the importance of learning from long-range time spans. Our model achieves state-of-the-art on several LVU tasks while being much more data efficient than previous works. Finally, we evaluate our model's transferability by setting a new state-of-the-art in five different benchmarks.

1. Introduction

Are movies just for entertainment? Arguably they offer much more than that. Movies are a source of inspiration for many and a force that influences societal behaviors [1]. Movies are also an active topic of study in the computer vision community [22, 13, 46, 5, 19, 3]. They have served as a testbed for measuring progress in visual recognition [16, 6], reasoning [36, 42], and creative editing [10, 31]. Moreover, movie data has been also leveraged to train computer vision [9], machine listening [7], and NLP [54] systems. Besides entertainment, movies are surely an intriguing media that can help AI models to understand semantics and artistic expressions encoded in a movie style.

While the quest of understanding movies has gained steady attention, it is still an open question how to develop models that leverage abundant sources of movie data [19, 37] to tackle all the movie-related tasks that the community has grown over the last decade [5, 26, 48, 41, 3].

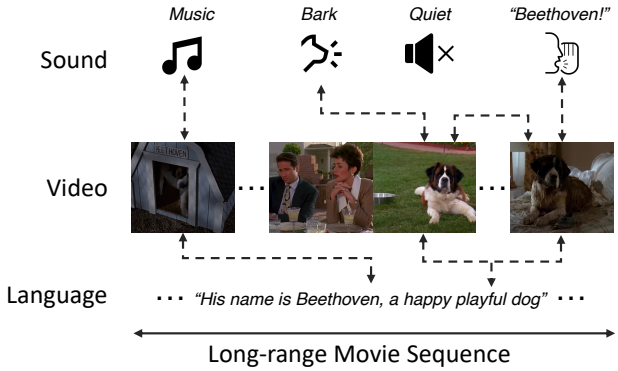

Figure 1. **Learning from Movie Sequences.** Movie sequences offer learning signals when observed for long-ranges. By reading the language (dialogue), hearing the sounds, and looking at the video, one can determine that Beethoven, the dog, was barking and he sleeps in a white and blue wooden dog house.

But, learning from movies is not a straightforward task, especially when no labels are available.

Existing video self-supervised approaches [11, 14, 2, 50, 52], which primarily focus on learning from short clips, would not leverage the richness of movies, as the value of movies as training sources emerges from their long-range dependencies. Moreover, the end-to-end learning scheme adopted in these works can not be easily extended to movies as it is computationally infeasible to encode long-form sequences in an end-to-end manner.

Recent works on long-form video understanding [48, 20, 9] have attempted to address these limitations by encoding movie clips using frozen base encoders [48, 9] or state-space models [20]. However, these works exclusively focus on the video modality for long-range temporal reasoning, disregarding audio and text signals, and hence are limited to specific tasks. In this work, we argue that long-form videos such as movies are rich in visual, auditory, and textual information, and integrating these modalities would lead to a better and generalizable understanding.

Fig. 1 illustrates how the three core modalities in movies (language, video, sound) can serve as a valuable source of supervision when reasoned all together for a long time. By analyzing the language (from the dialogue) we can understand that the dog's name is Beethoven, from the audio we could hear he is barking, and from long-range associations, we can infer he sleeps in a wooden dog house. This toy example illustrates how critical is to design training strategies that can effectively encode visual, audio, and language relationships. It is equally important to effectively design models that encode long-range dependencies.

This work presents pretraining strategy that leverages multimodal cues and long-range reasoning in movies. We develop a flexible model that can be easily transferred to a range of tasks related to movie analysis and understanding [48, 26, 3, 41, 5]. Our *first* design requirement is that the model needs to observe over a long time span in order to do long-range reasoning. To facilitate this, we dissect a long video into shots and represent it as an ordered sequence of shots. The main motivation for using shot-based input sampling instead of uniform temporal sampling is to capture longer multi-shot content at a time since each shot will be considered as one token irrespective of its length. Our *second* design requirement is to efficiently encode the input sequence by harnessing all available modalities. We do so by representing each shot clip in the sequence in video, audio, and language modalities. As it is computationally inefficient to encode long-form sequences in an end-to-end manner, we opt for using pretrained state-of-the-art models [44, 18, 4, 24] as base encoders to transform the raw video, audio, and text sequences into their corresponding compact feature representations.

Given a sequence of encoded base features for each modality, our *third* design requirement is to perform multimodal long-term reasoning in a self-supervised manner. To do so, we make use of Transformer networks [45]. Instead of combining all tokens from the different modalities as one long sequence like in VideoBERT [39], we follow a *hierarchical* approach. We first learn long-range context from each modality using *contextual* transformers while simultaneously ensuring that the context learned over one modality is also conditioned by another modality. We then learn joint representations between modalities using a *cross-modal* transformer network. To ensure that different transformers in our framework serve their purpose, we introduce a pretraining strategy that enforces intra-modal, inter-modal, and cross-modal relationships over long-range observations via carefully designed losses.

We train our model using publicly available movie dataset [37], performing additional preprocessing such as shot boundary detection [38]. We evaluate the transferability of our approach on six different benchmarks [48, 26, 3, 41, 5], and empirically show that long-range multimodal pretraining provides extensive benefits in performance.

Contributions. Our goal is to train transferable models for movie understanding. It brings two contributions:
(1) We introduce a pretraining strategy designed to leverage long-range multimodal cues in movies. We propose a model that captures intra-modal, inter-modal, and cross-modal dependencies via transformer encoders and self-supervision.
(2) We conduct extensive experiments to validate the transferability of our model and the contributions of the pretraining strategy. The results show that our model consistently improves the state-of-the-art across six benchmarks.

2. Related Works

Multimodal Pretraining in Long Videos. Pretraining from long videos has recently gained lots of attention [52, 40, 2, 50]. Previous works have built upon the intuition that modeling multimodality in long videos is key for learning. For instance, Zellers *et al.* [52] introduced an end-to-end BERT-alike model [12] that learns from sound, video, and transcripts from narrated YouTube videos. Similarly, [2] introduced multimodal models that use contrastive objectives to train base encoders on a large-scale collection of videos. While these approaches can offer specialized encoders for general perception tasks, their design choice for training the base encoders end-to-end prevents them to encode longer sequences and effectively modeling long-range dependencies. Concurrent to our work, Sun *et al.* introduced a framework for video-language pretraining in long videos [40]. Their proposed method shares some design choices with ours, such as including both video-language contrastive objectives, but also a cross-modal module trained with mask language modeling objectives [12]. Our method differs from these approaches in three key aspects: (i) we opt for leveraging frozen base encoders to facilitate longer modeling; (ii) our cross-modal objective is anchored on reconstructing the video representations from both audio and language modalities; and (iii) our method is specialized to movies and is pretrained using a much smaller dataset.

Learning from Movies. Movies have served as a popular source for training computer vision models [22, 47, 35, 51]. They have facilitated the creation of multiple datasets, enabling early research in action recognition [22], substantial progress in character recognition [13], and studies in cinematography [51, 3, 8], among many other areas [55, 5, 41]. Besides, movies have been also leveraged to train self-supervised models for various applications [30, 32, 33]. For example, Learning to Cut [30] recommends the best moments to cut a pair of shots by looking at motion. Finally, and closer to ours, recent methods have harnessed movies to train video representations for movies [21, 49, 20, 9]. Different from our approach, these works focus on train-

ing unimodal (visual) representations [49, 9, 20] or do not model long-range sequences [21].

Movie Understanding. There has been a growing interest in understanding long-form videos such as movies and TV shows with AI. To facilitate research in this direction, several datasets and benchmarks have been proposed. Huang *et al.* [19] proposed MovieNet, a dataset that contains 1,100 full movies and 60K trailers with annotations for various tasks including scene segmentation, character recognition, and genre prediction. Wu *et al.* [48] introduced the LVU benchmark which contains a total of 9 tasks related to content understanding, metadata prediction, and user engagement. Liu *et al.* [26] collected a large-scale dataset from TV shows for multi-shot temporal event localization. Some other works proposed [5, 41] movie clips-based datasets for text-video and audio-video retrieval tasks. Pardo *et al.* [31] proposed the Moviecuts dataset for studying different cutting patterns in movies. Recently, Argaw *et al.* [3] introduced the AVE benchmark for AI-assisted video editing. Our work offers a transferable model that can be tasked to address various movie understanding tasks.

3. Long-range Multimodal Pretraining

Given a movie M, we first dissect M into a sequence of shots using a shot-boundary detector [38]. Then, we sample k consecutive shots *i.e.* $\{S_1, S_2, \ldots, S_k\}$, as an input to our method. Our motivation for such a design choice comes in twofold. First, shot-based sampling captures longer multi-shot content at a time compared to uniform temporal sampling, since each shot is considered as one token irrespective of its length. Second, a model trained in this manner could be easily deployed to video editing tasks [3], where shots from different camera setups are sequentially assembled to make an edited scene. Each shot clip in the input sequence is represented in video (V), audio (A), and language (L) modalities as shown in Fig. 2. The video and audio inputs are directly extracted from the given shot sequence. For language input, we use all the text data spanning between the first and the last shot in the sequence.

3.1. Base Feature Encoding

As it is computationally inefficient to encode long sequences in an end-to-end manner, we opt for using state-of-the-art pretrained networks as base encoders. For the video inputs, $\{V_1, V_2, \ldots, V_k\}$, the corresponding base video features, *i.e.* $\{v_1, v_2, \ldots, v_k\}$, are obtained by passing each shot clip in the sequence into a frozen video backbone. In the same manner, base audio features, *i.e.* $\{a_1, a_2, \ldots, a_k\}$, are encoded from the input audio sequence using an audio backbone network. The base language features, *i.e.* $\{l_1, l_2, \ldots, l_s\}$ where s denotes the total number of word tokens, are extracted by encoding the raw text data using a pretrained language model. The goal of this step is to capture local information from each source and compress the captured information into a set of tokens to be used for the next steps. After base feature encoding, we project the encoded tokens across different modalities into a matching dimension using a linear layer.

3.2. Long-term Sequence Reasoning

Given a sequence of base features, we aim to design a model that integrates all available modalities in a self-supervised manner to perform long-term reasoning. First, we feed the base feature sequence from each modality into Transformer encoders [45] as shown in Fig. 2. The purpose of these transformers is to learn temporal context from the long-range input as each element in the given sequence attends to every other element in the sequence. Thus, we refer to them as *contextual* transformers. To ensure that the contextual transformers serve their purpose, we impose the following two constraints. First, we adopt a BERT-like [12] pretraining, where we replace 20% of the tokens with a special MASK token for audio, video, and text sequences. Each transformer is then trained to match the MASKed prediction with the corresponding input representation via intra-modality masking loss. For simplicity, let $\hat{w}$ denote the predicted representation for a masked input token w in a batch of $\mathcal{W}$ tokens. The *intra-modal* masking loss is then formulated as minimizing the cross entropy between $\hat{w}$ and w in all modalities as shown in Eq. (1) and Eq. (2).

$$\mathcal{L}_{\text{mask}}(\cdot) = \frac{1}{|\mathcal{W}|} \sum_{w \in \mathcal{W}} \left(\log \frac{\exp(\hat{w} \cdot w)}{\sum_{w \in \mathcal{W}} \exp(\hat{w} \cdot w)} \right) \quad (1)$$

$$\mathcal{L}_{\text{intra-modal}} = \mathcal{L}_{\text{mask}}(A) + \mathcal{L}_{\text{mask}}(V) + \mathcal{L}_{\text{mask}}(L) \quad (2)$$

While it is crucial to learn long-range contexts in each modality, it is equally important to ensure that the context learned over one modality is also conditioned by another modality in order to effectively capture the underlying multimodal cues in long-form videos. For instance, the soundtrack used in a movie often matches the visual content (*e.g.* pace, scene) of the movie. To enforce this notion in our model, we impose an inter-modal alignment between learned representations from the contextual transformers, *i.e.* contextual features. Using video modality as an anchor, we define the *inter-modal* contrastive loss as the sum of the InfoNCE loss [29] between audio $\leftrightarrow$ video, and video $\leftrightarrow$ language representations. Given the one-to-one alignment between contextual audio and video features ($\{\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_k\}$ and $\{\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_k\}$), we encourage the corresponding (positive) pairs to have high similarity and non-corresponding (negative) pairs to have low similarity as formulated in Eq. (3).

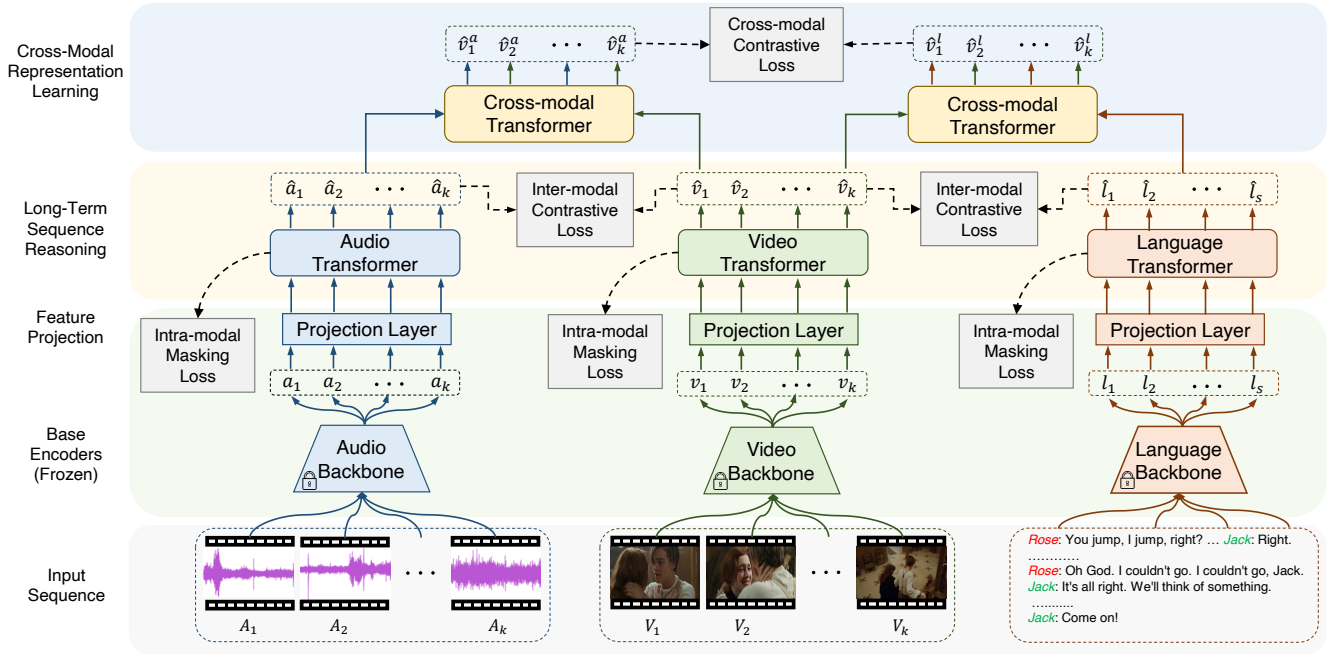

Figure 2. **Long-range Multimodal Pretraining.** Our approach takes as input audio, video, and language observations extracted from a sequence of k consecutive movie shots. The observations from each modality are encoded with *frozen* base encoders, and its outputs are projected via projection layers. Then, a stack of transformer encoders contextualize the base features of each modality. Finally, a cross-modal transformer is used to learn joint embeddings between the contextualized features. Our pretraining includes intra-modal, inter-modal, and cross-modal losses to jointly train the transformer models. The pretrained model can be used as a backbone encoder for several downstream tasks.

$$\mathcal{L}_{\hat{v}\to\hat{a}} = -\sum_{j}^{N}\sum_{i}^{k}\left(\log\frac{\exp(\mathcal{S}(\hat{v}_{j,i},\hat{a}_{j,i})/\tau)}{\sum_{j}^{N}\sum_{i}^{k}\exp\left(\mathcal{S}(\hat{v}_{j,i},\hat{a}_{j,i})/\tau\right)}\right) \quad (3)$$

, where j and i index batch size N and input sequence length k, respectively, $\mathcal{S}$ denotes cosine similarity and τ is a temperature parameter. The contrastive loss between contextual video and language features ($\{\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_k\}$ and $\{\hat{l}_1, \hat{l}_2, \ldots, \hat{l}_s\}$) is implemented using a one-to-many alignment scheme, *i.e.* for each video feature $\hat{v}_i$, the corresponding pair is obtained by averaging the representations of the language tokens spanning the interval of $\hat{v}_i$ such that:

$$\bar{l}_i = \frac{1}{C_i}\sum_{c=1}^{C_i}\hat{l}_c \quad (4)$$

, where C_i represents the number of language tokens in the interval of shot S_i. This is equivalent to matching each shot S_i to its corresponding textual representation. The resulting loss is then formulated similarly to the loss in Eq. (3) using $\hat{v}_i$ and $\bar{l}_i$. The total inter-modal contrastive loss is computed symmetrically as follows.

$$\mathcal{L}_{\text{inter-modal}} = \mathcal{L}_{\hat{v}\to\hat{a}} + \mathcal{L}_{\hat{a}\to\hat{v}} + \mathcal{L}_{\hat{v}\to\bar{l}} + \mathcal{L}_{\bar{l}\to\hat{v}} \quad (5)$$

3.3. Cross-modal Representation Learning

Thus far, our model learns to reason over long-term sequences in different modalities while simultaneously exploring their correlation. Intuitively speaking, the losses defined in Eq. (2) and Eq. (5) should be strong enough constraints to learn transferable representations. However, the interaction between the different modalities is enforced only at a loss level so far. Inspired by the success of joint encoding in related works [39, 2], we adopt a cross-modal Transformer [45] network to explicitly facilitate joint representation learning in our framework. The key motivation here is to further exploit multimodal cues from audio and language features to guide long-form video representation learning. We accomplish this by feeding the contextual audio/language features as a *source* sequence into the encoder, and the contextual video features as a *target* sequence into the decoder of the cross-modal transformer as shown in Fig. 2. Thus, our model decodes audio-conditioned and language-conditioned visual features, *i.e.* $\{\hat{v}_1^a, \hat{v}_2^a, \ldots, \hat{v}_k^a\}$ and $\{\hat{v}_1^l, \hat{v}_2^l, \ldots, \hat{v}_k^l\}$, by encoding the audio and language representations, respectively. To ensure that the cross-modal transformer serves its purpose, we introduce a cross-modal contrastive loss that encourages alignment between the learned cross-modal representations. It is defined as the InfoNCE loss [29] between audio-

Figure 3. Example of the input sequence of shots taken from the movie '101 Dalmatians (1996)'. The corresponding raw language data is composed of temporally aligned texts from both the subtitle data (*where there is speech*) and transcribed audio descriptions [37] (*where there is no speech*).

conditioned ($\{\hat{v}^a_1, \hat{v}^a_2, \ldots, \hat{v}^a_k\}$) and language conditioned ($\{\hat{v}^l_1, \hat{v}^l_2, \ldots, \hat{v}^l_k\}$) visual features as shown in Eq. (6).

$$\mathcal{L}_{\text{cross-modal}} = \mathcal{L}_{\hat{v}^a \rightarrow \hat{v}^l} + \mathcal{L}_{\hat{v}^l \rightarrow \hat{v}^a} \tag{6}$$

While the base feature encoders stay frozen, the remaining modules in the proposed framework are end-to-end trained by jointly optimizing the losses in Eq. (2), Eq. (5) and Eq. (6). Therefore, the total pretraining loss to self-supervise our model is defined by:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{intra-modal}} + \mathcal{L}_{\text{inter-modal}} + \mathcal{L}_{\text{cross-modal}} \tag{7}$$

Pretraining Dataset. We use the Movie Audio Description (MAD) [37] dataset to train our model. The dataset contains a diverse set of movies making more than 1200 hours of content. To ensure that every segment in a given movie has a corresponding language modality[1], we collect the official subtitle data for each movie and temporally align them with the textual descriptions from MAD as shown in Fig. 3. We prepare our training data by extracting video, audio, and language modalities from each full-length movie. For *video* modality, we use a shot-boundary detector [38] to segment a movie into a sequence of shot clips. We use FFmpeg to extract the corresponding *audio* of each shot clip. For *language* modality, we utilize the text data, including temporally aligned subtitles and audio descriptions, extracted from the beginning of the first shot to the end of the last shot in the sequence, while excluding overlapping content. We made sure that there is no overlap between the pretraining list and any of the videos (movies) in the test set of all downstream tasks by comparing IMDb ids.

Implementation Details. We set the shot sequence length $k = 30$ during pretraining. In other words, we use a 90-second movie clip as an input, since the shots in the pretraining data are approximately 3 seconds long on average. We use R(2+1)D-ResNet50 [44] model as the base video encoder. For the audio signals, we concatenate features from VGGish [18] and wav2vec 2.0 [4] networks; these features encode soundtrack and speech content representations, respectively. For the language input, we use the BART [24] model to encode the raw text data to base language features. We adopt a 3-layer Transformer architecture [45] for all contextual and cross-modal transformers. We use AdamW [27] optimizer, a cosine learning rate annealing strategy [43] with an initial value of $1e-3$ and a batch size of 1024 during pretraining. For all the InfoNCE [29] losses defined in Eq. (5) and Eq. (6) and we use a temperature parameter $\tau = 0.3$.

[1]MAD provides textual data only for movie segments with no speech.

4. Experiment

In this section, we first present ablations (Sec. 4.1) and experimental analyses (Sec. 4.2) on the widely used long-form video understanding (LVU) benchmark [48]. We then show the versatility of our approach (Sec. 4.3) in other five movie benchmarks related to event localization [26], scene understanding [3], editing pattern prediction [3], soundtrack selection [41] and scene description retrieval [5].

LVU Benchmark. The long-form video understanding (LVU) benchmark [48] contains 9 tasks that cover various aspects of long-form videos, including content understanding (*character relationship*, *speaking style*, *scene/place*), metadata prediction (*director*, *genre*, *writer*, *release year*), and user engagement regression (*like ratio*, *views*). It contains ∼11K videos, where each video is typically one-to-three minutes long. We set up the LVU tasks as follows. First, we use our pretrained model as an encoder to extract multimodal features from a given video. Second, we time-average the extracted sequence of features in order to get an aggregated representation for the input video. Third, we train a *single* classifier/regression layer on top of the time-

Table 1. **Ablation study** on the different losses during pretraining

Pretrain	$\mathcal{L}_{\text{intra}}$	$\mathcal{L}_{\text{inter}}$	$\mathcal{L}_{\text{cross}}$	Content ↑	Metadata ↑	User ↓
✗	✗	✗	✗	44.87	32.47	3.19
✓	✗	✓	✓	60.13	55.92	1.23
✓	✓	✗	✓	52.74	44.86	1.82
✓	✓	✓	✗	59.36	54.71	1.34
✓	✓	✓	✓	**60.53**	**57.67**	**1.16**

averaged feature for the different tasks in LVU.

4.1. Ablation Studies

In Table 1, we study the significance of each component in the proposed framework by pretraining our model in different settings and using the pretrained models as backbone encoders for downstream tasks. We report the average top-1 accuracy for content understanding and metadata prediction tasks, whereas average mean-squared error is used to evaluate user engagement tasks.

Pretraining. To show the importance of the pretraining step, we train the proposed model directly on downstream tasks from scratch. This resulted in poor performance compared to pretrained models as can be seen from Table 1. This is mainly because the tasks in LVU relatively have a small number of training samples which causes the model to overfit. More importantly, the classification/regression losses are not strong enough constraints to learn multimodal reasoning from long-form inputs.

Intra-modal Masking. Here, we analyze the benefit of the intra-modal contrastive loss ($\mathcal{L}_{\text{intra-modal}}$) for long-term sequence reasoning (Sec. 3.2). We do so by training our network without masking. As can be inferred Table 1, a model pretrained without $\mathcal{L}_{\text{intra-modal}}$ still gives a good performance on downstream tasks. This is intuitive because the inter-modal contrastive loss ($\mathcal{L}_{\text{inter-modal}}$) should implicitly guide each contextual transformer to reason over its input sequence, *i.e.* the model will effectively align the contextual representations only when it properly learns the context over each modality first. However, explicitly adding $\mathcal{L}_{\text{intra-modal}}$ during pretraining gave a performance boost as shown in Table 1.

Inter-modal Alignment. To examine the importance of imposing inter-modality alignment over the representations learned by contextual transformers (Sec. 3.2), we train our model without $\mathcal{L}_{\text{inter-modal}}$. A model pretrained in this manner gives a notably worse performance. This is most likely because the long-term sequence reasoning will be performed independently without inferring the intricacies between different modalities. This in turn affects the cross-modal representation learning (Sec. 3.3) as the cross-modal transformer will take a shortcut by ignoring audio and language representations. Thus, the learned representations will not be robustly transferable to downstream tasks explaining the poor performance in Table 1. These results reaffirm that inter-modality alignment is a critical loss for long-range multimodal pretraining.

Table 2. **Ablation study** on sequence length during pretraining

Sequence length (k)	Content ↑	Metadata ↑	User ↓
$k = 10$	59.77	56.45	1.20
$k = 30$ (Baseline)	60.53	57.67	1.16
$k = 60$	61.36	58.12	1.14

Table 3. **Analysis** on the contribution of different features

Features	Content ↑	Metadata ↑	User ↓
v_{base}	48.43	45.40	2.01
v_{base} + a_{base}	51.03	51.20	1.91
$\hat{v}_{\text{context}}$	56.50	53.68	1.82
$\hat{v}_{\text{context}}$ + $\hat{a}_{\text{context}}$	59.43	55.75	1.39
$\hat{v}_{\text{context}}$ + $\hat{a}_{\text{context}}$ + $\hat{v}^{a}_{\text{cross}}$	**60.53**	**57.67**	**1.16**

Cross-modal Transformer. We analyze the contribution of joint representation learning by training our model without the cross-modal transformer, *i.e.* without $\mathcal{L}_{\text{cross-modal}}$. As discussed in Sec. 3.3, optimizing $\mathcal{L}_{\text{intra-modal}}$ and $\mathcal{L}_{\text{inter-modal}}$ together should be a sufficient constraint to learn multimodal cues from long-form inputs. The empirical results in Table 1 also confirm this notion, where a model pretrained with $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{intra-modal}} + \mathcal{L}_{\text{inter-modal}}$ gives a competitive performance on downstream tasks. However, explicitly learning joint representation using the cross-modal transformer resulted in better performance.

Sequence Length. In Table 2, we study the effect of input sequence length during pretraining. To do so, we train 3 different models by setting the sequence length k to 10, 30, and 60. This translates to using 30, 90, and 180-second videos as input. As can be noticed from Table 2, there exists a pattern where a model pretrained with longer sequences gives better performance compared to a model pretrained with shorter sequences. This is intuitive because pretraining by observing longer sequences not only facilities the mining of more multimodal cues but also makes the model robust to various input video lengths at inference time.

4.2. Experimental Analyses

Learned Representations. Given our pretrained model as a backbone encoder, we study the contribution of the different representations (features) and their combination toward downstream task performance. Table 3 summarizes the results on LVU benchmark [48]. We combine features by simply concatenating (+) their time-averaged representation before feeding them to the classification/regression layer. As can be seen from Table 3, directly using base

Table 4. **Comparison with state-of-the-art methods** on long-form video understanding (LVU) benchmark.

Method	Content Understanding ↑				Metadata Prediction ↑					User Engagement ↓		
	Relationship	Way-Speaking	Scene	Average	Director	Genre	Writer	Year	Average	Like	Views	Average
SlowFast R101 [15]	52.4	35.8	54.7	47.6	44.9	53.0	36.3	**52.5**	46.7	0.386	3.77	2.08
VideoBERT [39]	52.8	37.9	54.9	48.5	47.3	51.9	38.5	36.1	43.4	0.320	4.46	2.39
CLIP [34]	56.1	36.7	52.9	48.6	56.2	50.9	37.8	46.4	47.8	0.411	3.85	2.13
Object Transformer [48]	53.1	39.4	56.9	49.8	51.2	54.6	34.5	39.1	44.8	0.230	3.55	1.89
ViS4mer [20]	57.1	40.8	67.4	55.1	62.6	54.7	48.8	44.7	52.7	0.260	3.63	1.95
Movie2Scenes [9]	67.7	**44.9**	63.8	58.8	**65.1**	57.5	**56.2**	51.8	57.6	**0.153**	2.46	1.31
LF-VILA [40]	61.5	41.3	**68.0**	56.9	-	-	-	-	-	-	-	-
Ours	**69.4**	44.4	67.8	**60.5**	64.9	**57.7**	55.8	52.3	**57.7**	0.163	**2.15**	**1.16**

video (v_{base}) and base audio (a_{base}) features results in a subpar performance. This is mainly because the frozen backbones used to encode the base features (Sec. 3.1) are specialized for capturing local information from short segments, and hence are ineffective for long-form inputs. In contrast, the video and audio features encoded from the contextual transformers ($\hat{v}_{\text{context}}$ and $\hat{a}_{\text{context}}$) give a significantly better performance as shown in Table 3. This confirms the benefit of long-term sequence reasoning discussed in Sec. 3.2. We experimentally observed that superior results on downstream tasks are obtained when features from both contextual and cross-modal transformers are combined. For instance, it can be inferred from Table 3 that the best-performing model on the LVU benchmark is $\hat{v}_{\text{context}} + \hat{a}_{\text{context}} + \hat{v}^{a}_{\text{cross}}$, where $\hat{v}^{a}_{\text{cross}}$ denotes the audio-conditioned visual feature.

Comparison with State-of-the-Art. In Table 4, we comprehensively compare our approach with state-of-the-art methods [15, 39, 34, 48, 20, 9, 40] on the LVU benchmark. As can be noticed from the table, short-term models such as VideoBERT [39] and CLIP [34] generally struggle to perform well in contrast to the long-term models such as ViS4mer [20], Movie2Scenes [9] or Ours. ViS4mer [20] proposes a transformer-based model with a state-space decoder for long video classification and trains a unique model from scratch for each task in LVU. In comparison, we only train a single linear layer, on top of the representations encoded from our pretrained model, for each task. As shown in Table 4, our model significantly outperforms ViS4mer [20] across different tasks despite having much fewer trainable parameters. These results highlight the strong merit of the proposed pretraining strategy which enabled a robustly transferable model for various long-form video understanding tasks.

Movie2Scenes [9] is pretrained using 30,340 movies and their associated metadata from Amazon Prime Video's *internal* database. [9] adopts a transformer network to learn video representation using movie similarity as a supervision signal. The comparison in Table 4 shows that our method gives a competitive if not better performance despite being pretrained on a dataset that is approximately 50 times smaller than the dataset used in Movie2Scenes. This is most likely attributed to the inter-modal and cross-modal constraints which enforce our model to learn the underlying multimodal cues between different modalities *within* a movie, thereby compensating for what it lacks in size. For instance, the genre of a movie can be inferred from audio signals, *e.g. explosion* sounds are usually associated with action-themed movies, while *unsettling* background music is usually used in horror movies. This explains the strong performance of our model in metadata prediction tasks in Table 4 even if it was pretrained without metadata information. In content understanding tasks, our method outperforms Movie2Scenes by 3% on average.

Table 5. **Temporal event localization** on MUSES benchmark.

Method	$\text{mAP}_{0.3}$	$\text{mAP}_{0.4}$	$\text{mAP}_{0.5}$	$\text{mAP}_{0.6}$	$\text{mAP}_{0.7}$	mAP
Random	1.20	0.64	0.29	0.10	0.03	0.45
P-GCN [53]	19.9	17.1	13.1	9.7	5.4	13.0
Liu *et al.* [26]	26.5	23.1	19.7	14.8	9.5	18.7
Ours	**29.5**	**26.9**	**22.8**	**16.6**	**13.3**	**21.8**

4.3. Transferable Model for Movie Understanding

In this section, we study the capabilities of our pretrained model beyond long-form video classification. For this purpose, we make use of five movie benchmarks related to event localization [26], editing [3], and retrieval [41, 5]. For each benchmark task, we follow the official data split and training (fine-tuning) settings.

Multi-shot Temporal Event Localization. Localizing events in movies is a very challenging task due to the large intra-instance variation caused by frequent shot cuts. This includes actor/scene/camera change and heavy occlusions within a single instance. We thus explore the potential of our work for temporal event localization in multi-shot videos. For this purpose, we use the multi-shot events (MUSES) benchmark [26]. MUSES is an action localization dataset with 25 categories and contains 3,697 videos collected from more than 1000 drama episodes. Each video in the dataset is typically five-to-nineteen minutes long.

We follow the task pipeline used by the state-of-the-art

Table 6. **Cinematographic scene understanding** on AVE.

Method	Shot Size	Shot Angle	Shot Type	Shot Motion	Shot Location	Shot Subject	Num People	Sound Source	Average
CLIP [34]	51.3	54.9	58.0	37.6	81.0	42.9	57.2	43.4	53.3
ResNet101 [17]	66.8	55.9	64.7	33.5	82.1	46.8	60.2	32.0	55.2
AVE [3]	65.0	49.5	**65.3**	43.2	83.7	46.7	61.4	38.9	56.7
Ours	**67.4**	**57.7**	63.8	**46.1**	**84.2**	**51.2**	**61.8**	**50.1**	**60.3**

method of Liu *et al.* [26]. Given a video and a set of temporal proposals, the overall process in [26] consists of three steps, *i.e.* feature extraction, temporal aggregation, and proposal evaluation. The key difference between [26] and our approach is the temporal aggregation step. While [26] trains a convolution-based temporal aggregation module to mitigate the intra-instance variation, we instead use the features encoded from the transformers in our pretrained model.

We use the mean average precision (mAP) metric to evaluate the performance of multi-shot event localization. In Table 5, we compare our approach and state-of-the-art methods [53, 26] using different threshold values of temporal intersection over union (IoU). As can be inferred from Table 5, using our pretrained model as an encoder for the temporal aggregation step leads to a significantly better localization performance compared to the baseline model [26]. For example, our approach achieves an average mAP of 21.8 on MUSES which is approximately 16.6% better than the state-of-the-art method. This is mainly because our pretrained model is capable of reasoning over a multi-shot input, and hence the localization network can focus on the proposal evaluation step unlike [26] where the network also has to learn temporal aggregation from scratch.

Cinematographic Scene Understanding. In the filmmaking process, movie scenes are created by sequentially assembling shots. Thus, it is apparent that movie scene understanding should be formulated by taking the building blocks, *i.e.* shots, into account. In this regard, cinematographic scene understanding aims at predicting the attributes of the shots that compose a given scene. To evaluate the transferability of our pretraining to this task, we use the anatomy of video editing (AVE) benchmark [3]. AVE introduces eight tasks related to cinematographic attributes prediction, *i.e.* *shot size*, *shot angle*, *shot type*, *shot motion*, *shot location*, *shot subject*, *number of people*, *sound source*. The dataset contains a total of 196,176 shots obtained from 5,591 publicly-available movie scenes.

While AVE [3] formulates shot attributes classification problem on an individual shot basis, we perform a scene-level attributes prediction instead, where we first encode a scene (shot sequence) using our pretrained model and then train a *single* classifier layer on top of the encoded sequence to simultaneously predict a specific attribute for each shot in the scene. Table 6 shows the comparison of our approach and competing baselines on cinematographic scene understanding. Following [3], we use the average per-class accuracy metric for evaluation. As can be inferred from Table 6, the shot sequence representations encoded from our pretrained model result in better performance in different tasks. These results are achieved by only training classifier layers, while previous methods fine-tune their whole network for specific tasks.

Table 7. **Editing pattern prediction** on AVE.

Shot Sequence Ordering		Next Shot Selection	
Method	Acc.	Method	Acc.
Random	16.6	Random	20.0
Argaw *et al.* [3] (late fusion)	24.4	Cosine similarity	13.4
Argaw *et al.* [3] (early fusion)	30.7	Argaw *et al.* [3]	41.4
Ours	**32.5**	Ours	**44.2**

Editing Pattern Prediction. Here we evaluate the proposed pretraining scheme for two tasks related to video editing, *i.e.* *shot sequence ordering* and *next shot selection*. Shot sequence ordering (SSO) is defined as a classification problem where a network is tasked to predict the order of shots given a sequence of contagious but randomly shuffled shots. Next shot selection (NSS), on the other hand, takes a partial sequence of shots as a context and predicts the most-likely next shot from a list of candidate shots. We use the AVE benchmark[3], which formulates SSO as a 6-way classification task with 3 shots at a time, and NSS as a multiple-choice problem with a sequence of 9 shots, where the first 4 shots are used as a context and the remaining 5 make the candidate list.

We follow the task setup used by [3]. Given a sequence of shots, [3] extracts audio-visual features from each shot clip and concatenates the extracted features as input to the next layer. In our case, we pass the extracted sequence of features into our pretrained contextual and cross-modal transformers before concatenating them. This simple modification notably improves network performance for both shot sequence ordering and next shot selection tasks. These results indicate that the proposed pretrained framework is capable of reasoning over long sequence of videos and can capture aspects related to movie style, enabling potential applications in automated video editing.

Scene-Soundtrack Selection. Scene-soundtrack selection aims to retrieve a soundtrack that best matches the

Table 8. **Scene-soundtrack selection**

Method	Median Rank ↓		Recall ↑		
	V → A	A → V	R@1	R@5	R@10
Random	1000	1000	0.05	0.25	0.50
MVPt [41]	21	21	15.03	25.74	36.56
Ours (zero-Shot)	33	29	12.54	23.03	33.61
Ours (fine-tuned)	**13**	**10**	**15.72**	**34.96**	**46.80**

Table 9. **Scene description retrieval** on CDM benchmark

Method	R@1	R@5	R@10	Median R.	Mean R.
Random	0.01	0.08	0.15	3290	3290.0
CDM [5]	5.6	17.6	26.1	50	243.9
Ours (zero-shot)	5.4	17.3	25.2	56	254.7
Ours (fine-tuned)	**7.7**	**23.1**	**32.8**	**27**	**176.3**

theme of a given scene and vice versa. To evaluate the performance of our approach on this task, we use the recently proposed MovieClips-based benchmark [41]. Given a scene (soundtrack), we follow [41] and compute the feature distance between each soundtrack (scene) in a pool of $N = 2000$ target candidates *not seen during model pretraining*. After sorting the candidates based on the distance value, we evaluate the retrieval performance using two metrics: (i). *Recall@K* checks if the ground-truth pair is in the K closest candidates and, (ii). *Median Rank* returns the median value of the positions of the ground-truth pair in the sorted list of candidates across the test set.

Table 8 shows the comparison of our work with competing baselines. The results indicate that performing zero-shot video $\rightleftharpoons$ audio retrieval using the contextual audio and video features already gives competitive results. The inferior performance compared to the state-of-the-art method, MVPt [41], can be attributed to the fact that our base audio encoders (VGGish [18] and wav2vec 2.0 [4]) are not well-suited for extracting discriminative music features, thereby making zero-shot retrieval a challenging task. We address this limitation by fine-tuning (on [41]) our pretrained model using DeepSim [23], which extracts disentangled music tagging embeddings, as our base audio encoder, *i.e.* we only fine-tune the contextual video and audio transformers using inter-modal contrastive loss in Eq. (3). Our fine-tuned model achieves a new state-of-the-art performance on scene-soundtrack selection. To examine the contribution of the pretraining step in the achieved result, we train the audio and video transformers from scratch using the fine-tuning dataset. We experimentally observed that such training achieves a performance equivalent to MVPt's [41]. This confirms the importance of the pretrained model for video $\rightleftharpoons$ audio retrieval task.

Scene Description Retrieval. Given a high-level semantic description of a scene, this task targets to retrieve the correct scene over all possible candidates in a dataset. To evaluate our model on this task, we use the Condensed movies (CDM) benchmark [5] which contains a total of 33,976 scenes and their corresponding textual description. At test time, a feature embedding from a given text query is compared with 6,581 scene embeddings out of which only one is the correct pair. We compare the performance of our approach and several state-of-the-art methods [25, 28, 5] in different metrics as summarized in Table 9. For zero-shot scene description retrieval, we use the outputs of the contextual video and language transformers.

As can be seen from Table 9, our pretrained model gives a competitive performance compared to previous approaches, even though it hasn't been specifically optimized for this task. We also experimented with fine-tuning the video and language branches of our model using the CDM [5] train set. Our fine-tuned model outperforms the previous state-of-the-art, CDM [5], by a significant margin. The experimental results in Table 8 and Table 9 highlight the flexibility of the proposed approach, where different components of our framework can be either deployed directly or further fine-tuned for specific tasks.

5. Conclusion

We introduced a new pretraining strategy that leverages multimodal cues over long-range videos. We designed a model that incorporates contextual transformer layers for each modality (audio, video, language) and cross-modal transformers to capture long-range dependencies in movie clips. We trained the model in a modest dataset of movies to later test it on six different movie understanding benchmarks. Our extensive experimental results empirically demonstrated the effectiveness of our long-range multimodal pretraining strategy.

Acknowledgement. This work was supported by the Institute of Information and communications Technology Planning and evaluation (IITP) grant funded by the Korea government (MSIT, No. 2021-0-02068, Artificial Intelligence Innovation Hub) and National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT, No. RS-2023-00212845).

References

[1] https://www.nytimes.com/interactive/2018/11/30/movies/women-in-movies.html. 1

[2] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *Advances in Neural Information Processing Systems*, 34:24206–24221, 2021. 1, 2, 4

[3] Dawit Mureja Argaw, Fabian Caba Heilbron, Joon-Young Lee, Markus Woodson, and In So Kweon. The anatomy of

video editing: A dataset and benchmark suite for ai-assisted video editing. In *The European Conference on Computer Vision (ECCV)*, 2022. 1, 2, 3, 5, 7, 8

[4] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460, 2020. 2, 5, 9

[5] Max Bain, Arsha Nagrani, Andrew Brown, and Andrew Zisserman. Condensed movies: Story based retrieval with contextual embeddings, 2020. 1, 2, 3, 5, 7, 9

[6] Digbalay Bose, Rajat Hebbar, Krishna Somandepalli, Haoyang Zhang, Yin Cui, Kree Cole-McLaughlin, Huisheng Wang, and Shrikanth Narayanan. Movieclip: Visual scene recognition in movies. *arXiv preprint arXiv:2210.11065*, 2022. 1

[7] Sourish Chaudhuri, Joseph Roth, Daniel PW Ellis, Andrew Gallagher, Liat Kaver, Radhika Marvin, Caroline Pantofaru, Nathan Reale, Loretta Guarino Reid, Kevin Wilson, et al. Ava-speech: A densely labeled dataset of speech activity in movies. *arXiv preprint arXiv:1808.00606*, 2018. 1

[8] Boris Chen, Amir Ziai, Rebecca Tucker, and Yuchen Xie. Match cutting: Finding cuts with smooth visual transitions. *arXiv preprint arXiv:2210.05766*, 2022. 2

[9] Shixing Chen, Xiang Hao, Xiaohan Nie, and Raffay Hamid. Movies2scenes: Learning scene representations using movie similarities. *arXiv preprint arXiv:2202.10650*, 2022. 1, 2, 3, 7

[10] Robin Courant, Christophe Lino, Marc Christie, and Vicky Kalogeiton. High-level features for movie style understanding. In *ICCV 2021 Workshop on AI for Creative Video Editing and Understanding*, 2021. 1

[11] Ishan Dave, Rohit Gupta, Mamshad Nayeem Rizve, and Mubarak Shah. Tclr: Temporal contrastive learning for video representation. *Computer Vision and Image Understanding*, 219:103406, 2022. 1

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2, 3

[13] Mark Everingham, Josef Sivic, and Andrew Zisserman. Hello! my name is... buffy"–automatic naming of characters in tv video. In *BMVC*, volume 2, page 6, 2006. 1, 2

[14] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. *arXiv preprint arXiv:2205.09113*, 2022. 1

[15] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 7

[16] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6047–6056, 2018. 1

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 8

[18] 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 *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE, 2017. 2, 5, 9

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

[20] Md Mohaiminul Islam and Gedas Bertasius. Long movie clip classification with state-space video models. *arXiv preprint arXiv:2204.01692*, 2022. 1, 2, 3, 7

[21] Mahdi M Kalayeh, Nagendra Kamath, Lingyi Liu, and Ashok Chandrashekar. Watching too much television is good: Self-supervised audio-visual representation learning from movies and tv shows. *arXiv preprint arXiv:2106.08513*, 2021. 2, 3

[22] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 1, 2

[23] Jongpil Lee, Nicholas J Bryan, Justin Salamon, Zeyu Jin, and Juhan Nam. Metric learning vs classification for disentangled music representation learning. *arXiv preprint arXiv:2008.03729*, 2020. 9

[24] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019. 2, 5

[25] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1871–1880, 2019. 9

[26] Xiaolong Liu, Yao Hu, Song Bai, Fei Ding, Xiang Bai, and Philip HS Torr. Multi-shot temporal event localization: a benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12596–12606, 2021. 1, 2, 3, 5, 7, 8

[27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[28] Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a text-video embedding from incomplete and heterogeneous data. *arXiv preprint arXiv:1804.02516*, 2018. 9

[29] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3, 4, 5

[30] Alejandro Pardo, Fabian Caba, Juan León Alcázar, Ali K Thabet, and Bernard Ghanem. Learning to cut by watching movies. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6858–6868, 2021. 2

[31] Alejandro Pardo, Fabian Caba Heilbron, Juan León Alcázar, Ali Thabet, and Bernard Ghanem. Moviecuts: A new dataset and benchmark for cut type recognition. *arXiv preprint arXiv:2109.05569*, 2021. 1, 3

[32] Georgios Pavlakos, Jitendra Malik, and Angjoo Kanazawa. Human mesh recovery from multiple shots. In *CVPR*, 2022. 2

[33] Georgios Pavlakos, Ethan Weber, Matthew Tancik, and Angjoo Kanazawa. The one where they reconstructed 3d humans and environments in tv shows. In *European Conference on Computer Vision*, pages 732–749. Springer, 2022. 2

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 7, 8

[35] Anyi Rao, Jiaze Wang, Linning Xu, Xuekun Jiang, Qingqiu Huang, Bolei Zhou, and Dahua Lin. A unified framework for shot type classification based on subject centric lens. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 17–34. Springer, 2020. 2

[36] Arka Sadhu, Tanmay Gupta, Mark Yatskar, Ram Nevatia, and Aniruddha Kembhavi. Visual semantic role labeling for video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5589–5600, 2021. 1

[37] Mattia Soldan, Alejandro Pardo, Juan León Alcázar, Fabian Caba, Chen Zhao, Silvio Giancola, and Bernard Ghanem. Mad: A scalable dataset for language grounding in videos from movie audio descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5026–5035, 2022. 1, 2, 5

[38] Tomáš Souček and Jakub Lokoč. Transnet v2: An effective deep network architecture for fast shot transition detection. *arXiv preprint arXiv:2008.04838*, 2020. 2, 3, 5

[39] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473, 2019. 2, 4, 7

[40] Yuchong Sun, Hongwei Xue, Ruihua Song, Bei Liu, Huan Yang, and Jianlong Fu. Long-form video-language pre-training with multimodal temporal contrastive learning. *arXiv preprint arXiv:2210.06031*, 2022. 2, 7

[41] Dídac Surís, Carl Vondrick, Bryan Russell, and Justin Salamon. It's time for artistic correspondence in music and video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10564–10574, 2022. 1, 2, 3, 5, 7, 9

[42] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4631–4640, 2016. 1

[43] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 5

[44] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 2, 5

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 3, 4, 5

[46] Paul Vicol, Makarand Tapaswi, Lluis Castrejon, and Sanja Fidler. Moviegraphs: Towards understanding human-centric situations from videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8581–8590, 2018. 1

[47] Paul Vicol, Makarand Tapaswi, Lluis Castrejon, and Sanja Fidler. Moviegraphs: Towards understanding human-centric situations from videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[48] Chao-Yuan Wu and Philipp Krahenbuhl. Towards long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1884–1894, 2021. 1, 2, 3, 5, 6, 7

[49] Fanyi Xiao, Kaustav Kundu, Joseph Tighe, and Davide Modolo. Hierarchical self-supervised representation learning for movie understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9727–9736, 2022. 2, 3

[50] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *arXiv preprint arXiv:2109.14084*, 2021. 1, 2

[51] Min Xu, Jinqiao Wang, Muhammad A Hasan, Xiangjian He, Changsheng Xu, Hanqing Lu, and Jesse S Jin. Using context saliency for movie shot classification. In *2011 18th IEEE International Conference on Image Processing*, pages 3653–3656. IEEE, 2011. 2

[52] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. Merlot reserve: Neural script knowledge through vision and language and sound. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16375–16387, 2022. 1, 2

[53] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7094–7103, 2019. 7, 8

[54] Yinhe Zheng, Rongsheng Zhang, Minlie Huang, and Xiaoxi Mao. A pre-training based personalized dialogue generation model with persona-sparse data. In *Proceedings of the*

AAAI Conference on Artificial Intelligence, volume 34, pages 9693–9700, 2020. 1

[55] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015. 2