Hierarchical Self-supervised Representation Learning for Movie Understanding

Fanyi Xiao*, Kaustav Kundu, Joseph Tighe, Davide Modolo
AWS AI Labs
{kaustavk,tighej,dmodolo}@amazon.com

Abstract

Most self-supervised video representation learning approaches focus on action recognition. In contrast, in this paper we focus on self-supervised video learning for movie understanding and propose a novel hierarchical self-supervised pretraining strategy that separately pretrains each level of our hierarchical movie understanding model (based on [37]). Specifically, we propose to pretrain the low-level video backbone using a contrastive learning objective, while pretrain the higher-level video contextualizer using an event mask prediction task, which enables the usage of different data sources for pretraining different levels of the hierarchy. We first show that our self-supervised pretraining strategies are effective and lead to improved performance on all tasks and metrics on VidSitu benchmark [37] (e.g., *improving on semantic role prediction from 47% to 61% CIDEr scores). We further demonstrate the effectiveness of our contextualized event features on LVU tasks [54], both when used alone and when combined with instance features, showing their complementarity.*

1. Introduction

Most of the latest research on self-supervised video representation learning (SSL) focuses on the task of action recognition [4, 9, 13, 17, 32, 34, 55]. This priority has largely influenced the design of these methods, as well as the type of datasets used to learn their representations. For example, they propose models that encourage the learning of short-term appearance and motion cues, as these are the most informative for action recognition. At the same time, they mostly focus on pretraining on the Kinetics [20] dataset, which consists of hundreds of thousands of short YouTube clips with diverse motion and semantic patterns. Unlike these works, we are interested in learning self-supervised video representations to understand movies.

Movies are however very complex and they require reasoning at many levels: from the simple understanding of low-level actions to the interpretation of high-level semantic narratives, which require knowledge of the characters, their histories, relationships, behaviours, etc. Towards building rich models for movie understanding, [37] recently proposed a hierarchical movie understanding model that learns in a fully supervised fashion. However, it is extremely difficult to annotate large-scale video datasets, even for a relatively simple task like action classification, not to mention for complex movie tasks (*e.g.* labeling actor relation graphs [45]). To overcome this bottleneck, we propose a novel hierarchical self-supervised pretraining strategy that separately pretrains each level of this hierarchical model.

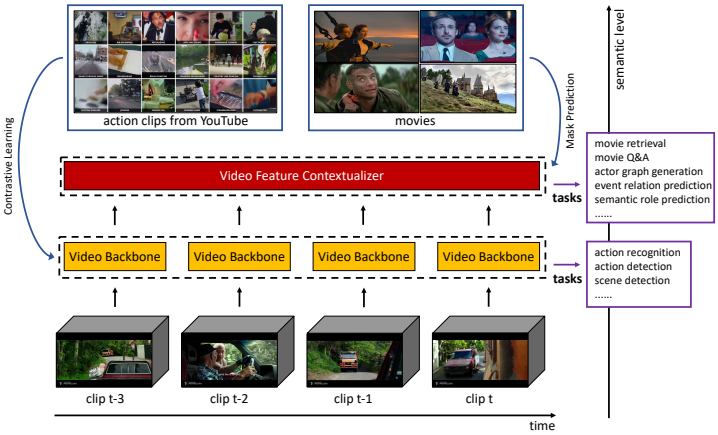

Figure 1. ***Hierarchical self-supervised pretraining***. *We pretrain the low-level video feature backbone using contrastive learning objectives on large collections of YouTube-style action clips; while pretrain the higher-level feature contextualizer using mask prediction on movies with rich temporal plots.*

In details, the hierarchical movie model of [37] consists of two levels: a low-level video backbone encoder and a higher-level transformer contextualizer (Fig. 1). We design our hierarchical learning strategy to sequentially pretrain the backbone and the transformer encoder as they specialize in different aspects of movie understanding. The backbone is responsible for the heavy-lifting work to extract low-level appearance and motion cues for people, objects and scenes from raw pixels. Therefore, it needs to be high in capacity and can be trained on a large amount of YouTube videos (e.g., Kinetics [21]). Once we obtain an appropriate feature abstraction from the video backbone, we can treat such rep-

*Work done while at Amazon, now at Meta AI

resentations as visual "word tokens" and learn to contextualize the neighboring visual tokens. The contextualizer can be lightweight and trained on a small amount of training data with stronger semantic and temporal structures (i.e., movies).

In details, we propose to pretrain the video backbone using a *contrastive learning* objective, which helps models learning the intra-instance invariances from visual cues. This pretraining paradigm has shown to be very effective for action recognition [13, 17, 32, 34, 55]. Furthermore, we pretrain the higher level transformer model to produce contextualized semantic representations using the *mask prediction* task, which has been shown effective for pretraining language models that take in word tokens for contextualization [7, 25]. These hierarchical self-supervised pretraining strategies bring two data advantages: they enable the use of different data sources for pretraining the different levels of the hierarchy, and they do not require any annotation, which are inherently expensive to collect.

We evaluate the impact of our pretrainings on the recently released VidSitu [37] and LVU [54] datasets. These are movie datasets that have been annotated for various tasks, ranging from low-level verb prediction (i.e., actions) to high-level semantic role prediction or event relation classification (i.e., "A causes B"). Our results show that our self-supervised pretraining strategies are effective and lead to improved performance on all tasks and metrics. For example, on the task of Semantic Role Prediction, we improve CIDEr [44] metric performance over the previous, fully-supervised, state-of-the-art [37] from 47% to 61%. Finally, we also ablate the design choices of our pretraining recipes.

2. Related Work

Self-supervised video representation learning. Many works have explored ways to learn representations by designing pretext tasks that exploit the temporal structure of videos. For example, some works attempted to learn representations by predicting the ordering of video frames [14, 27], while others instead designed the task of predicting direction [51] and speed [4] of the video. Others attempted to learn video representations by either tracking across frames patches [48], pixels [50], colors [46], or by predicting temporal context for videos [9, 34, 47]. A more recent line of work overcomes the need for pretext tasks by leveraging the *contrastive learning* paradigm [13, 17, 32, 55] and achieved impressive results even when comparing to fully-supervised methods. Though flourishing, all of the works mentioned above focus on learning video representations from short YouTube-style action clips (*e.g.*, Kinetics) and have action recognition as the task in mind when designing learning objectives and architectures. In contrast, we are interested in learning video representations *from movies* and *for movies*, which as we will elaborate in next sections, require very different learning objectives and architectures.

From this perspective, [38, 41, 54] are closest to our work in that they also pretrain a transformer for feature contextualization. However, [41, 54] focus on masking spatial regions, either the object boxes in [54] or small patches in [41], to learn the spatial arrangements in videos. Whereas [38] relies on joint video and language masking, which requires aligned video-narration pairs. In contrast, we demonstrate we can directly pretrain our contextualizer with mask prediction using a simple event-level representation, without needing any object detectors trained with box supervision, or video-text pairs.

Movie understanding. Researchers have explored many individual movie understanding tasks, including low-level tasks like spatio-temporal action detection [16], scene detection [19], metadata classification (*e.g.*, genre) [54], as well as tasks that require higher-level contextualization and reasoning like movie description [35], movie question answering [42], story-based retrieval [2], semantic role prediction [37] and social graph generation [45]. Unlike these work which mostly focus on a single task, we demonstrate the general benefits of our pretraining strategy by transferring it to a hierarchy of movie tasks.

Contextualized temporal modeling for videos. One distinct characteristic of movie understanding is that there exists strong semantic correlation across neighboring scenes and events [29,42]. An effective way to learn temporal contextualization is to apply RNNs to model the evolution of frames [10,22,23,30,40]. To deal with longer temporal window, it's helpful to establish an explicit feature bank to store useful features along time [53]. To model finer-grained interactions, there are works that leverage pre-computed object/person proposals or detections [3, 26, 39, 49]. Though related, our focus is different in that we're interested in developing effective pretraining methods for feature contextualization, and also we want to achieve so without using external object or person detectors for preprocessing.

3. Hierarchical SSL for Movies

Understanding movies is a complex task and it requires reasoning at many levels. Towards learning rich representations for movies, [37] was the first work to propose a hierarchical model that consists of a low-level CNN video feature backbone and a high-level transformer feature contextualizer. Inspired by it, this work aims at delving deeper into hierarchical movie understanding and explore the importance of pretraining for movies. Specifically, we re-evaluate the choices of [37] (Sec. 3.1), which pretrained their video backbone on a fully supervised action dataset and trained the contextualizer from scratch, and demonstrate that our pretraining strategies are better designed to help each level of the hierarchy learn features that are meaningful for the task of movie understanding. Since each level is responsible

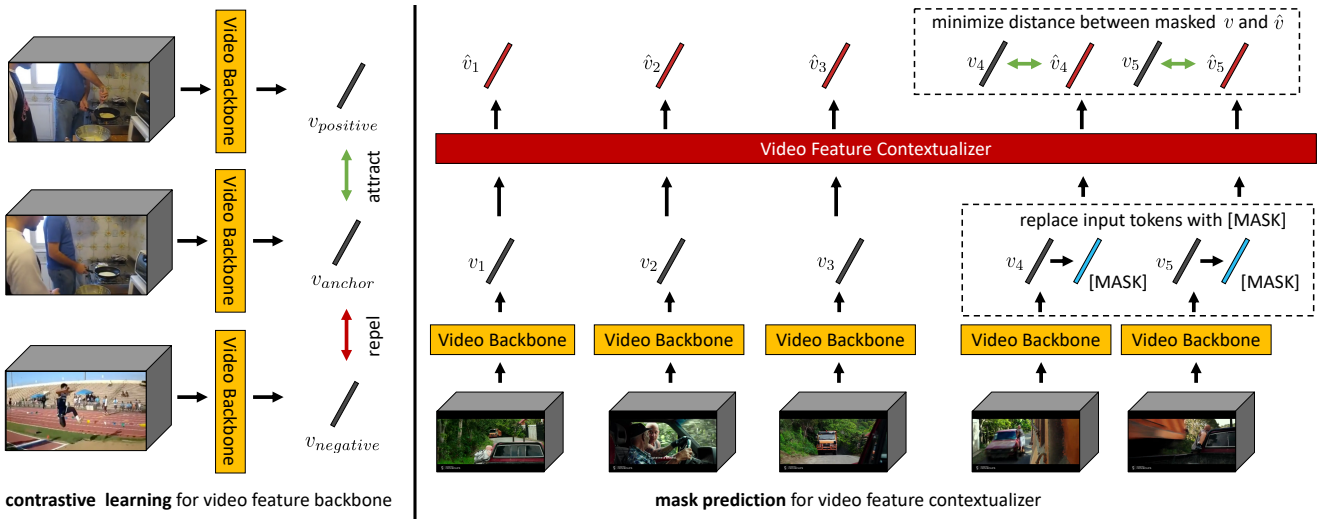

Figure 2. ***Overview of our hierarchical pretraining methods***. *The **left** shows how we use contrastive learning to pretrain our video feature backbone — features v_{anchor} and $v_{positive}$ produced from two clips of the same video are pulled together to each other, whereas the feature $v_{negative}$, computed from a clip sampled from another video, is pushed away. Whereas the **right** shows how we pretrain our feature contextualizer using mask prediction — in this sequence of 5 tokens, we mask out input tokens v_2 and v_3 to the contextualizer, and then forward through to get the outputs $\hat{v}_i$. We then set the learning objective to minimize the distance between the output tokens ($\hat{v}_2$, $\hat{v}_3$) and the masked-out input tokens (v_2, v_3).*

for different goals, we propose to separate the pretraining of the video backbone and the feature contextualizer.

As the video backbone is responsible for extracting low-level appearance and motion cues, we propose to pretrain it using self-supervised contrastive learning (Sec. 3.2), which explicitly models the intra-instance invariances. On the other hand, the video feature contextualizer is responsible for propagating information across neighboring clips ("visual tokens"). Inspired by the NLP literature [8], we propose to pretrain it using an event-level mask prediction task (Sec. 3.3), with some key differences to how it's applied in previous approaches. In contrast to some recent works in computer vision that propose to apply mask predictions to learn *spatial arrangements* of video patches or objects [41, 54], we focus on learning the *temporal contextualization* for event representations. Furthermore, unlike [38], which requires joint video and language masking, we show that our method learns strong contextualized event features using only video clips. Since our method does not need any object detector [54] or synced video-narration pairs [38], it makes our method simpler and more scalable.

Finally, this decouping also enables us to pretrain the different levels on different datasets. This leads to better specialization (since we can use the most suitable dataset for each level) and less reliance on having large-scale datasets for the target domain (i.e., we do not require hundreds of thousands of movies to train the expensive video backbone). This is an important advantage over previous SSL methods that mostly conduct pretraining of the full model with a single task and dataset.

3.1. Hierarchical model for movie understanding

We follow [37] and adopt their hierarchical architecture for movie understanding (Fig. 1). It uses a 3D CNN as the low-level video feature backbone and a transformer encoder and decoder for feature contextualization and natural language generation, respectively. The video backbone extracts features v_t on short 2-seconds clips. Then, the transformer encoder operates on a sequence $\{v_t, v_{t+1}, ..., v_N\}$ of consecutive clip features and contextualizes them into $\hat{v}_t$. These contextualized features are finally used as input to either a classifier (*e.g.* for visual tasks, like event relation prediction), or a transformer decoder (TxD), that decodes natural language outputs (*e.g.*, for semantic role prediction).

For the backbone, we adopt the popular Slow-only network [12], but with two modifications [32]: 1) instead of 8×8 inputs (8 frame inputs sampled 8 frames apart), we use denser 16×4 inputs with a temporal kernel of 5 for the first conv layer to increase its temporal receptive field; 2) the inputs are downsampled with a temporal stride of 2 after the first conv layer, to save computation. We denote this backbone as Slow-D for its denser inputs.

For the transformer encoder (TxE) and decoder architecture (TxD), we largely follow [37]. Specifically, for the transformer encoder, we use 3 layers of multi-head self-attention with residual connections, with 16 heads each and a hidden dimension of 1024. For inputs, we append a learned position embedding to each of N tokens in the input sequence, which we found to work better compared to the sinusoidal embedding used in [37]. The transformer decoder also has 3 layers, each one consisting of a self-

attention module and a cross-attention module, where the self-attention is computed for text inputs only, while the cross-attention is added for the text tokens to query visual tokens as keys. For more details on the architecture, please refer to our supp. materials.

3.2. Video backbone: contrastive pretraining

We adopt instance discrimination contrastive learning, as it has been demonstrated to be very effective to learn visual semantics patterns by capturing the intra-instance invariances [6, 18, 32, 34, 55]. Among these, we experiment with two simple, yet robust, methods: CVRL [32] and MoDist [55]. CVRL pretrains using the popular InfoNCE objective [31]. Their goal is to pull together the representations of two clips sampled from the same video, while pushing apart those of clips that are sampled from different videos (Fig. 2 left). Though it yields impressive results, CVRL does not explicitly utilize motion cues for representation learning. MoDist [55] addresses this with a visual-motion cross-modal contrastive objective where a supporting motion network is used to distill information to the visual backbone, so that it can learn motion-sensitive features.

3.3. Contextualizer: mask prediction pretraining

The goal of pretraining the transformer is to make it better at contextualizing the individual semantic tokens (video clips in our settings). For this, we use the mask prediction task, which is widely used in language model training in NLP (*e.g.*, BERT [8]). Specifically, as shown in Fig. 2 (right), given a set of visual tokens $\{v_1, v_2, ..., v_N\}$, we randomly select a mask size m from the set $m \in \{1, 2, ..., \alpha N\}$, where $\alpha \in [0, 1]$ determines the ratio of the max mask size with respect to the sequence length, and a mask starting position $s \in \{1, 2, ..., N - m + 1\}$. With the sampled size and location, we mask out the selected tokens $\{v_s, ..., v_{s+m-1}\}$ and replace them with a special `[MASK]` token. Then, we forward the masked sequence through the transformer encoder to obtain its L_2 normalized outputs $\{\hat{v}_1, \hat{v}_2, ..., \hat{v}_N\}$. Ideally with proper contextualization from neighboring clips, even with the input clip masked out, $\hat{v}_t$ should still be able to "fill in" the semantic information carried in v_t (*e.g.*, predicting the "car passing by" scene for the last two input tokens in Fig. 2 right, given the first three tokens as contexts). Solving this problem is key to learning the rich temporal dynamic of movies and we formulate its learning objective in a way that pushes the output $\hat{v}_t$ to be close to its corresponding input v_t:

$$\mathcal{L}_{mp} = -\log \frac{\exp(\hat{v}_t \cdot v_t / \tau)}{\exp(\hat{v}_t \cdot v_t / \tau) + \sum_{i=1}^{K} \exp(\hat{v}_t \cdot p_i / \tau)}, \quad (1)$$

where τ is a temperature parameter, and p_i are a set of distractions, from which we would like $\hat{v}_t$ to identify v_t by predicting its semantics from contexts. We construct the pool of distractions $\{p_1, p_2, ..., p_K\}$ by maintaining a FIFO queue during training. Note that a simpler alternative is to directly enforce an L_2 loss between $\hat{v}_t$ and v_t, but empirically we found this to yield slightly worse results than Eq. 1 (Table 2). Finally, note that though possible, we do not study the pretraining of the transformer decoder in this work, as we are mainly interested in pretraining movie representations that can be transferred generically (i.e., backbone+TxE), whereas TxD is task-dependent (e.g., the decoder for SRL is a multimodal transformer that takes in both texts and videos) and is only used for certain tasks.

4. Experiments: VidSitu Benchmark

VidSitu [37] is a comprehensive movie understanding benchmark that features different tasks ranging from the low-level, visual-only "verb prediction" on short clips, to the higher-level, multimodal "semantic role prediction" and "event relation" classification. The dataset contains 29k 10-seconds clips from 3k different movies from MovieClips [1]. The dataset provides detailed annotations for each clip, including 1) verb class labels at 2-sec intervals (i.e., each 10-sec clip is split into 5 "events"), 2) semantic role labels for each annotated verb, and 3) labels specifying relations between two events (*e.g.*, event A `is caused by` event B). The dataset is split into a train set of 23.5k clips and a val set of 1.3k clips, on which we evaluate our models. Finally, to avoid data contamination, we remove 241 videos from VidSitu val that are overlapping with LVU dataset.

Implementation details. For self-supervised pretraining of our video backbone, we use the training sets of the large-scale Kinetics-400 [20] (K400, 240k clips) and the much smaller scale, but in-domain, VidSitu (23.5k). We pretrain both CVRL and MoDist for 400 epochs on K400 and, when specified, an additional 200 epochs on VidSitu. We pretrain our transformer encoder only on movie clips from VidSitu and LVU [54] (which is another dataset of 10k movie clips), as we want the transformer encoder to learn about temporal contextualization from movies. We pretrain TxE with mask prediction for 100 epochs on VidSitu and 1000 epochs on LVU as the movie clips are ~10x longer than VidSitu. Note that we never use any human labels from the datasets for either of these pretraining (backbone and TxE). For contextualizer hyperparameters, we set input sequence length as $N = 5$, $\alpha = 0.6$, $\tau = 0.1$, $K = 65536$. Please refer to our supp. materials for more details about pretraining.

4.1. Semantic role prediction

In this section, we study the effectiveness of self-supervised pretraining for semantic role prediction, which is a very challenging movie understanding task, due to its rich output space (free form natural language) and its multimodal nature (visual and language). The goal of this task

Model	pretraining backbone	pretraining TxE	CIDEr [44]	CIDEr-verb	CIDEr-arg	ROUGE-L [24]	LEA [28]
GPT2 [33]	-	-	34.67	42.97	34.45	40.08	48.08
I3D+TxD [37]	Sup-K400	-	47.14	51.61	41.29	40.67	37.89
I3D+TxE+TxD [37]	Sup-K400	N	47.06	51.67	42.76	42.41	48.92
Slow-D+TxD	Sup-K400	-	51.37 ± 1.06	59.68 ± 0.88	46.10 ± 0.90	41.37 ± 0.59	36.03 ± 0.70
Slow-D+TxE+TxD	Sup-K400	N	51.36 ± 1.04	59.72 ± 0.87	47.25 ± 0.94	41.72 ± 0.65	45.99 ± 0.56
Slow-D+TxE+TxD	CVRL-K400	N	54.40 ± 0.96	63.18 ± 1.27	47.63 ± 1.83	41.80 ± 1.01	46.31 ± 1.13
Slow-D+TxE+TxD	CVRL-K400	MaskPred-K400	57.48 ± 1.74	65.08 ± 1.92	51.21 ± 1.87	41.65 ± 1.15	45.71 ± 0.87
Slow-D+TxE+TxD	CVRL-VS	MaskPred-VS	44.32 ± 0.56	52.07 ± 0.81	39.08 ± 0.64	40.56 ± 0.34	**48.87 ± 0.62**
Slow-D+TxE+TxD	CVRL-K400	MaskPred-VS	60.34 ± 0.75	69.12 ± 1.43	53.87 ± 0.97	**43.77 ± 0.38**	46.77 ± 0.61
Slow-D+TxE+TxD	CVRL-K400	MaskPred-LVU	**61.18 ± 1.48**	**69.15 ± 1.57**	**54.99 ± 1.12**	43.38 ± 0.87	47.81 ± 0.90
Human (up. bound)			84.85	91.70	80.15	39.77	72.10

Table 1. ***Semantic role prediction results on VidSitu.*** *Results in the top section are taken from [37]. The bottom row shows the human performance by measuring the agreement between annotators [37], which serves as the performance upper bound. Unlike [37], which only reports results for a single run, we found there is large variance across runs (likely due to the fact that this task evaluates free-form natural language outputs), therefore we run 10 times for each experiment and report its mean and standard error.*

is to predict various semantic role labels for each verb, including for example the agent and the patient of the verb, as well as other attributes like the scene where the verb is happening, and the description about how it's carried out (e.g. "urgent"). Due to high human disagreement on some of the annotated roles, the benchmark only evaluates the agent (e.g. "person"), the patient (e.g. "ball"), the instrument/benefactive/attribute (e.g. "towards a basket") and the location/scene (see Sec. 4.1 of [37] for more details). Following [37], we evaluate using the CIDEr [44] score metric (and its variants CIDEr-verb and CIDEr-arg that are CIDEr scores averaged across verbs and argument types). Furthermore, for completeness we also report ROUGE-L [24] and LEA [28]. Finally, since we observe large variance across runs for this task, we run 10 times for each experiment and report its mean and standard error.

Impact of self-supervised pretraining. We present our results in Table 1. The top section presents results from [37]: `GPT2` is a visual-blind language model baseline that only takes in verb classes as input; `I3D+TxD` directly takes the I3D [5] video features as the input to the transformer decoder (TxD) without using the transformer encoder (TxE) to contextualize the features; and `I3D+TxD+TxE` adds the TxE contextualizer to it. The second section of the table presents our results of different pretraining settings for the video backbone and TxE. Throughout, we use 'N' to denote 'not pretrained'.

Several interesting observations arise from these results. As reported in [37], the TxE contextualizer does not bring any performance gain over the simpler `I3D+TxD` model (47.06 *vs*. 47.14 CIDEr) and our results using Slow-D show a similar trend (51.37 *vs*. 51.36). However, rather than drawing the conclusion that contextualization is not helpful in this case, we instead hypothesized that this is due to the lack of proper pretraining for TxE, and we could resolve this with our self-supervised pretraining using mask prediction on event features. Our results validate our intuition, as TxE pretraining significantly outperforms training TxE from scratch. The largest improvement comes from pretraining on the LVU dataset (which is 4.6x larger than VidSitu, therefore better performance), which improves CIDEr from 54.40 to 61.18. This large gain comes from the fact that the mask prediction task essentially forces TxE to learn to contextualize input tokens (*i.e*., event features in this case) by propagating useful information among them. While this has been demonstrated with large success in training language models like BERT, it has only been applied in vision in the form of predicting masked *spatial regions* like patches [41], objects boxes [54], or joint vision-language pretraining that requires video-narration pairs [38]. To the best of our knowledge, we are the first to show that this can be generalized to simple event-level video representations learned only using videos, and can lead to significant improvements over the state-of-the-art (61.18 *vs*. 47.14).

In Sec. 3 we discussed the importance of separating pretraining for the backbone and the contextualizer to enable specialized training on the most suitable datasets. To quantify the importance of selecting the right dataset for each pretraining, we evaluate all possible permutations of using K400 and VidSitu (e.g., "K400+VS" refers to a CVRL backbone pretrained on K400, followed by a TxE pretrained using mask prediction on VidSitu). Among these, K400+VS achieves the highest performance (60.34), showing the importance of learning the backbone on the largest-scale dataset, but the TxE on an in-domain (i.e., movie) one. Interestingly, "VS+VS" performs the worst by a large margin, showing how the backbone does not need in-domain data to learn low-level video features and can generalize well to movies when pretrained appropriately.

Finally, it is also promising to see that pretraining the backbone in a self-supervised fashion (CVRL-K400) generalizes better than fully supervised pretraining (Sup-K400): 54.40 *vs*. 51.36, which is likely due to the large gap between the supervised pretraining task and the downstream

mask size	stride	sampling	loss	CIDEr	CIDEr-verb	CIDEr-arg	ROUGE-L	LEA
{1}	2s	uniform	contrastive	57.01 ± 2.21	63.80 ± 3.06	50.85 ± 1.77	41.69 ± 1.34	**49.13 ± 1.24**
{1, 2}	2s	uniform	contrastive	59.15 ± 2.01	66.79 ± 1.99	53.56 ± 1.61	42.16 ± 1.13	46.33 ± 0.94
{1, 2, 3}	2s	uniform	contrastive	**61.18 ± 1.48**	**69.15 ± 1.57**	**55.00 ± 1.12**	**43.38 ± 0.87**	47.81 ± 0.90
{1, 2, 3, 4}	2s	uniform	contrastive	59.45 ± 1.31	64.71 ± 2.05	52.94 ± 1.55	43.25 ± 0.56	48.96 ± 0.61
{1, 2, 3}	1s	uniform	contrastive	60.15 ± 1.13	66.81 ± 1.64	53.29 ± 1.17	42.31 ± 0.68	47.66 ± 0.69
{1, 2, 3}	2s	uniform	contrastive	**61.18 ± 1.48**	**69.15 ± 1.57**	**55.00 ± 1.12**	**43.38 ± 0.87**	47.81 ± 0.90
{1, 2, 3}	3s	uniform	contrastive	58.38 ± 0.80	65.69 ± 0.72	52.15 ± 1.11	42.50 ± 0.49	**48.02 ± 0.70**
{1, 2, 3}	2s	uniform	L_2 distance	58.88 ± 1.13	66.43 ± 0.86	52.22 ± 1.01	43.05 ± 0.60	**48.55 ± 0.85**
{1, 2, 3}	2s	max discrep.	contrastive	**61.65 ± 0.79**	**68.44 ± 0.93**	**55.06 ± 0.90**	**43.44 ± 0.45**	48.25 ± 0.54

Table 2. ***Ablating mask prediction for TxE pretraining.*** *The **top** section ablates the effectiveness of different sizes of the mask to apply (*e.g.*, '{1, 2}' refers to sample mask sizes of 1 and 2). The **middle** section ablates the impact of different token strides (*e.g.*, '2s' refers to having two neighboring tokens as features computed from event segments that are 2s apart). The **bottom** section ablates alternative loss function and mask sampling strategy that adaptively select tokens to mask out.*

target task (action recognition *vs*. SRL in this case).

Ablating TxE pretraining. We now conduct ablation experiments to understand the impact of our design choices for TxE mask prediction pretraining (Table 2). We experiment with the best model from Table 1: [`Slow-D+TxE+TxD, CVRL-K400, MaskPred-LVU`]. First, we study the impact of varying the size of the mask in Table 2 (top). Among all the tested options, uniformly sampling mask sizes from {1, 2, 3} achieves the best performance. Using a higher or lower value decreases performance considerably. This is understandable, as it would be too challenging to predict 4 masked out tokens out of a total of 5 tokens, meanwhile it would be too easy when too few tokens are masked out.

Next, we study the impact of the stride size between two consecutive tokens (Table 2 middle). While all entries achieve competitive results, computing video features of events that are 2-seconds apart achieves the best balance.

Finally, we ablate on two other aspects of our mask prediction task in Table 2 (bottom): loss function and mask sampling strategy. First, we change the loss function from Eq. 1 to a standard L_2 loss. This lowers the accuracy, likely due to L_2 being more sensitive to the representation collapse issue [15]. Then, we compare the simple uniform sampling (used in all experiments) with a more sophisticated sampling strategy for mask positions: "max discrepancy". We believe that selecting good tokens to mask can help the model learn better. As an example, conceptually it would be more helpful for the model to learn to "fill in" a masked out event of a person showing painful expression from the preceding context event of another person punching with the fist, compared to masking out in the middle of a long shot of someone talking. Specifically, given a token v_i, discrepancy captures the difference in TxE output between $\hat{v}_i$ which is computed without any masking and $\hat{v}'_i$, which is computed by masking v_i. High discrepancy indicates that v_i is an important token for TxE and masking it will push TxE to learn harder, using only the remaining tokens. Towards this, "max discrepancy" samples the token with the highest discrepancy. Surprisingly, this sampling strategy only achieve slightly better CIDEr score compared to the simple uniform sampling. We hypothesize that this is likely due to the nature of VidSitu, which contains short movie highlights, rather than more temporally coherent full movies and so the majority of tokens are already challenging if masked.

4.2. Event relation prediction

We now study the effectiveness of self-supervised pretraining on the even relation prediction task. This task is formulated as a 4-way classification problem between four relation types: "A is enabled by B", "A is a reaction to B", "A causes B", and "A is unrelated to B". Annotations are provided as (A, B, relation) triplets. To predict the relation between events A and B, we concatenate features of both events either directly from the video backbone Slow-D (*i.e.*, v_A and v_B) or from the contextualizer TxE (*i.e.*, $\hat{v}_A$ and $\hat{v}_B$). We experiment with features from different models (Slow-D and Slow-D+TxE) and pretraining techniques in Table 3. Following [37], we evaluate our results using the mean accuracy (averaged over relation types) and top-1 accuracy on the provided validation set.

We present results for two settings: in the first one we directly transfer the pretrained features to event relation prediction (vb finetune '✗'). Whereas in the second setting, we take the models that are further finetuned on VidSitu verb prediction ('✓'), as done in [37]. With both, self-supervised pretraining can be as effective as supervised pretraining, especially when using the motion-sensitive MoDist features (33.29% *vs*. 33.03%, and 34.66% *vs*. 34.00%).

Interestingly, we found that it does not work when naively adding the TxE and training it from scratch using randomly initialized weights, in which case the model does not train and reaches chance performance (Mean-Acc 25%). This is likely why [37] directly used the features from the video backbone and disregarded the outputs of TxE for this task. However, our results show that with proper pretraining using mask prediction, one can train Slow-D+TxE model

model	backbone pretrain	vb finetune	TxE pretrain	Mean-Acc	Top1-Acc
I3D [37]	Supervised	✓	-	34.13	39.91
Slow-D	Supervised	✗	-	33.03 ± 0.21	41.90 ± 0.23
Slow-D	CVRL	✗	-	32.05 ± 0.24	38.68 ± 0.69
Slow-D	MoDist	✗	-	33.29 ± 0.15	40.52 ± 0.58
Slow-D	No	✓	-	30.66 ± 0.21	41.29 ± 0.31
Slow-D	Supervised	✓	-	34.00 ± 0.12	40.65 ± 0.29
Slow-D	CVRL	✓	-	33.89 ± 0.17	41.35 ± 0.24
Slow-D	MoDist	✓	-	34.66 ± 0.18	41.75 ± 0.47
Slow-D+TxE	CVRL	✓	No	25.00 ± 0.00	39.42 ± 0.00
Slow-D+TxE	MoDist	✓	No	25.00 ± 0.00	39.42 ± 0.00
Slow-D+TxE	CVRL	✓	MaskPred	34.71 ± 0.07	41.16 ± 0.24
Slow-D+TxE	MoDist	✓	MaskPred	**35.32 ± 0.17**	41.62 ± 0.43

Table 3. ***Event relation prediction on VidSitu.*** `Supervised`*: pretrained on K400 with class labels.* `CVRL`/`MoDist`*: pretrained with either CVRL [32] or MoDist [55] first on K400 and then on VidSitu, as we found it helps bridging domain gaps to movies. Methods in the* ***top*** *section are pretrained and then directly transferred to event relation prediction on VidSitu. In the* ***middle*** *section, methods are further finetuned on VidSitu verb prediction task (vb finetune '✓'). Finally, the* ***bottom*** *section shows the results of appending the transformer encoder (TxE) following the video backbone. For each experiment, we repeat 10 runs and report its mean and standard error.*

backbone	pretrain	Acc@1	Acc@5	Recall@5
Slow [37]	Sup-K400	29.05	58.69	19.19
Slow-D	-	31.69	68.64	5.68
Slow-D	Sup-K400	38.29	69.27	**18.70**
Slow-D	CVRL-K400	32.84	61.57	13.59
Slow-D	MoDist-K400	42.96	73.17	17.48
Slow-D	CVRL-K400+VS	35.29	65.92	14.41
Slow-D	MoDist-K400+VS	**44.67**	**74.38**	18.40

Table 4. ***Verb prediction results on VidSitu.*** *The first row shows the result applying the Slow-only network in [37]. We evaluate with top-1/5 accuracy as well as recall@5 metrics, following [37]. For pretrain settings, we use the 'method-dataset' notation.*

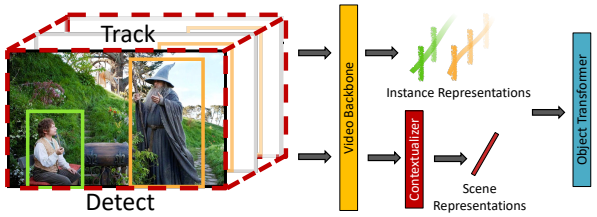

Figure 3. ***Object Transformer++.*** *Given the video and the detected/tracked objects, the top pathway feeds their cropped instance features into an Object Transformer to model their interactions [54]. In addition to this, as shown in the bottom pathway, we propose to add a scene-level event representation, produced by our pretrained contextualizer (Sec. 3.3), to model the context of the scene beyond the detected objects.*

for this task, and achieve better accuracy than using the backbone features alone: our mask prediction pretraining improves performance for both CVRL and MoDist, leading to state-of-the-art results of 35.32% mean-acc compared to previous best result of 34.13% reported in [37].

4.3. Verb prediction

Finally, we evaluate on the verb prediction task, which is the standard task of predicting action classes on short video segments. Each movie clip in the dataset is split into five 2-seconds event segments, each one annotated with a verb label. The dataset contains 1560 verb classes, like "look", "talk", "walk", "run", "grab", "drive", etc. We follow [37] and evaluate results using top-1 and top-5 accuracy (Acc@1/5) and top-5 recall (Rec@5) in Table 4. We observe the followings. (i) A self–supervised video backbone pretrained with MoDist outperforms a backbone pretrained with action labels on K400 (Acc@1: 42.96 *vs*. 38.29). This is surprising since some of the labels in this verb prediction task are the same as the action classes in K400. We argue this is due to the domain gap between YouTube-style action clips in K400 and movies in VidSitu, which self-supervised pretraining helps reducing. (ii) Between the two SSL methods, MoDist performs better than CVRL, suggesting its motion-sensitive features are well-suited for many verbs with strong motion like "walk" and "run". (iii) Finally, when we extend to pretrain on both K400 and VidSitu (K400+VS), the performance further improves, as VidSitu helps to reduce the domain gap. While this is expected, one should note that it is only possible due to the self-supervised nature of the pretraining. To achieve a similar benefit using fully supervised pretraining, one would have to annotate videos in the new domain (e.g., genres for VidSitu movies), which is however expensive and non-scalable.

5. Experiments: LVU Benchmark

In this section, we demonstrate the effectiveness of our approach on the Long-form Video Understanding (LVU) dataset [54]. LVU is a large-scale dataset of 10k videos (typically 1-3 minutes long) with 9 diverse tasks, including user engagement (*YouTube like ratio*, *popularity*), movie meta data classification (*director*, *genre*, *writer*, *movie release year*) and content understanding classification (*relationship of actors in the scene*, *speaking style*, *scene*).

instance	scene	Relation(↑)	Speaking(↑)	Scene(↑)	Director(↑)	Writer(↑)	Year(↑)	Genre(↑)	Like(↓)	Views(↓)	#top-1	mean
OT [54]†	✗	50.00	**34.57**	32.56	37.38	26.55	25.73	49.55	0.396	4.559	1/9	3.22
CVRL	✗	50.95	32.86	32.56	37.76	27.26	25.31	48.17	0.444	4.600	0/9	3.89
MoDist	✗	49.52	33.57	30.70	40.56	23.10	26.57	49.26	0.458	4.506	0/9	3.89
✗	Sup	**52.38**	34.37	26.51	23.18	5.36	18.88	47.37	0.595	**4.061**	2/9	4.44
✗	Ours	**52.38**	33.07	36.98	**42.43**	23.93	**35.24**	48.11	0.375	4.653	3/9	2.89
OT [54]	Ours	50.95	34.07	**44.19**	40.19	**31.43**	29.65	**51.15**	**0.353**	4.886	**4/9**	**2.33**

Table 5. ***LVU tasks.*** *The first two columns show the pretraining setting we use for both the instance and scene representations ('✗' denotes 'not used'). For each of the 9 tasks, we show the top-1 accuracy if it's a classification task, and mean squared error for regression tasks. In the last two columns, we show the number of tasks an approach gets the highest rank, and the mean rank of each approach across all tasks. †: note how the numbers of OT in the first row are lower than the ones reported in [54] because (a) we use a R50 instead of R101 backbone, (b) we use a Slow-D network for spatio-temporal features instead of the more expensive SlowFast network, and (c) we pretrain on 10k movies mentioned in their public repository instead of 30k movies used in their paper.*

Object Transformer++. In [54], the authors propose a long-term temporal model called Object Transformer (OT). It uses an object-centric design to represent each video as a set of spatio-temporal instances (i.e., tracklets of people and objects, Fig. 3 top pathway) and a transformer-based architecture [43] to model the synergies of the tracked instances in the video (blue rectangle). We argue that while such an object-centric design is useful for modeling long-term movie understanding, it is not sufficient. Reasoning only about the objects and the interaction among them can overlook the context of the scene, which is critical to understand movies (i.e., actors move out of the camera view, but the scene continues). Instead, we propose to enrich OT with a new scene representation. Specifically, we propose to use our self-supervised pretrained contextualizer (TxE, Fig. 3 bottom pathway) to supplement the instance features. We denote the enhanced full method in Fig. 3 as OT++.

Results. In Table 5, we report the results across all the 9 tasks proposed in the LVU dataset using the same parameters and experimentation protocol as [54]. We report the average performance over 5 runs. 'instance' and 'scene' are the two pathways of Fig. 3. The former denotes the object-centric features proposed by OT [54], while the latter denotes the context features we propose as a mean to improve OT. 'OT', 'CVRL' and 'MoDist' denote different features representing the instance tracklets ('instance' pathway). 'OT' is first pretrained with full-supervision on K400 and then with instance masking on LVU, exactly as in [54]. 'CVRL' and 'MoDist' instead are only pretrained with contrastive self-supervision on K400. On the other hand, 'Ours' and 'Sup' instead encode whole frames ('scene' pathway). 'Ours' denotes our self-supervised hierarchical pretraining, which consists of a Slow-D+TxE model with the backbone pretrained on K400 using CVRL, and TxE contextualizer pretrained with event mask prediction on LVU. While 'Sup' is only a Slow-D backbone trained fully supervised on K400 without any contextualization (*i.e.*, w/o the red block in Fig. 3 bottom pathway). Finally, we use all these previously mentioned encoders to embed video clips and feed them to a final Object Transformer (blue box in Fig. 3) that is finetuned for the 9 LVU tasks.

In the first three rows we report the performance of using only the instance representation pathway, as is in [54]. The first row shows results for the OT baseline [54]. As expected, features trained with only contrastive objectives (row 2-3) underperform [54], which shows the effectiveness of OT for long term movie understanding. However, when we use the contextualized features produced by pretraining with our event-level mask prediction task on LVU, even our scene representations w/o any instance features (Fig. 3, bottom pathway only) already outperform the much more complicated instance model OT (mean rank 2.89 *vs*. 3.22). When we combine this with the instance representation of OT, we obtain OT++, which achieves the best performance overall (mean rank 2.33), showing they are complementary. In addition, we also observe a significant improvement using our contextualized scene representations, compared to a scene representation directly pooled from the fully-supervised, but not contextualized, backbone feature (row 5 *vs*. 4, mean rank 2.89 *vs*. 4.44), showing the importance of our pretrained contextualizer. Finally, we note that 3 tasks experience a performance drop when we combine instance and scene representations, likely because these tasks (*e.g.*, predicting the year or director of a movie) do not rely on specific instance representations.

Conclusion

We proposed a novel hierarchical self-supervised pretraining method tailored for movie understanding. Specifically, we proposed to separately pretrain each level of our hierarchical movie understanding model, so that they can become experts within the relevant domain (e.g., learn low-level appearance and motion patterns vs high-level contextualization). We demonstrated the effectiveness of our pretraining strategies on both VidSitu and LVU benchmarks, achieving state-of-the-art results. We hope these strategies will serve as a first baseline and encourage new research towards self-supervised learning for movies. Finally, for this research we used the following public codebases: PySlowFast [11], VidSitu [36] and LVU toolkits [52].

References

[1] MovieClips. https://www.movieclips.com. 4

[2] Max Bain, Arsha Nagrani, Andrew Brown, and Andrew Zisserman. Condensed movies: Story based retrieval with contextual embeddings. In *ACCV*, 2020. 2

[3] Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *ECCV*, 2018. 2

[4] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. SpeedNet: Learning the speediness in videos. In *CVPR*, 2020. 1, 2

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 5

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 4

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 2

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 2019. 3, 4

[9] Ali Diba, Vivek Sharma, Luc Van Gool, and Rainer Stiefelhagen. DynamoNet: Dynamic action and motion network. In *ICCV*, 2019. 1, 2

[10] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015. 2

[11] Haoqi Fan, Yanghao Li, Bo Xiong, Wan-Yen Lo, and Christoph Feichtenhofer. PySlowFast GitHub Repository. https://github.com/facebookresearch/slowfast, 2020. 8

[12] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast Networks for Video Recognition. In *ICCV*, 2019. 3

[13] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. In *CVPR*, 2021. 1, 2

[14] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *CVPR*, 2017. 2

[15] 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. In *NeurIPS*, 2020. 6

[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 *CVPR*, 2018. 2

[17] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *NeurIPS*, 2020. 1, 2

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 4

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

[20] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1, 4

[21] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1

[22] Dong Li, Zhaofan Qiu, Qi Dai, Ting Yao, and Tao Mei. Recurrent tubelet proposal and recognition networks for action detection. In *ECCV*, 2018. 2

[23] Zhenyang Li, Kirill Gavrilyuk, Efstratios Gavves, Mihir Jain, and Cees GM Snoek. Videolstm convolves, attends and flows for action recognition. *Computer Vision and Image Understanding*, 2018. 2

[24] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 2004. 5

[25] 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 preprint arXiv:1907.11692*, 2019. 2

[26] Chih-Yao Ma, Asim Kadav, Iain Melvin, Zsolt Kira, Ghassan AlRegib, and Hans Peter Graf. Attend and interact: Higher-order object interactions for video understanding. In *CVPR*, 2018. 2

[27] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 2016. 2

[28] Nafise Sadat Moosavi and Michael Strube. Which coreference evaluation metric do you trust? a proposal for a link-based entity aware metric. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016. 5

[29] Seil Na, Sangho Lee, Jisung Kim, and Gunhee Kim. A read-write memory network for movie story understanding. In *ICCV*, 2017. 2

[30] J. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015. 2

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

[32] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *CVPR*, 2021. 1, 2, 3, 4, 7

[33] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019. 5

[34] Adrià Recasens, Pauline Luc, Jean-Baptiste Alayrac, Luyu Wang, Florian Strub, Corentin Tallec, Mateusz Malinowski, Viorica Patraucean, Florent Altché, Michal Valko, et al. Broaden your views for self-supervised video learning. *arXiv preprint arXiv:2103.16559*, 2021. 1, 2, 4

[35] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description. In *CVPR*, 2015. 2

[36] Arka Sadhu. VidSitu GitHub Repository. https://github.com/TheShadow29/VidSitu, 2021. 8

[37] Arka Sadhu, Tanmay Gupta, Mark Yatskar, Ram Nevatia, and Aniruddha Kembhavi. Visual semantic role labeling for video understanding. In *CVPR*, 2021. 1, 2, 3, 4, 5, 6, 7

[38] 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, 3, 5

[39] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *ECCV*, 2018. 2

[40] Lin Sun, Kui Jia, Kevin Chen, Dit-Yan Yeung, Bertram E Shi, and Silvio Savarese. Lattice long short-term memory for human action recognition. In *ICCV*, 2017. 2

[41] Hao Tan, Jie Lei, Thomas Wolf, and Mohit Bansal. VIMPAC: Video Pre-Training via Masked Token Prediction and Contrastive Learning. *arXiv preprint arXiv:2106.11250*, 2021. 2, 3, 5

[42] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *CVPR*, 2016. 2

[43] 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. 8

[44] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015. 2, 5

[45] Paul Vicol, Makarand Tapaswi, Lluis Castrejon, and Sanja Fidler. Moviegraphs: Towards understanding human-centric situations from videos. In *CVPR*, 2018. 1, 2

[46] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *ECCV*, 2018. 2

[47] Jue Wang, Gedas Bertasius, Du Tran, and Lorenzo Torresani. Long-short temporal contrastive learning of video transformers. *arXiv preprint arXiv:2106.09212*, 2021. 2

[48] Xiaolong Wang and Abhinav Gupta. Unsupervised Learning of Visual Representations using Videos. In *ICCV*, 2015. 2

[49] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, 2018. 2

[50] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019. 2

[51] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *CVPR*, 2018. 2

[52] Chao-Yuan Wu. LVU GitHub Repository. https://github.com/chaoyuaw/lvu, 2021. 8

[53] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *CVPR*, 2019. 2

[54] Chao-Yuan Wu and Philipp Krahenbuhl. Towards long-form video understanding. In *CVPR*, 2021. 1, 2, 3, 4, 5, 7, 8

[55] Fanyi Xiao, Joseph Tighe, and Davide Modolo. Modist: Motion distillation for self-supervised video representation learning. *arXiv preprint arXiv:2106.09703*, 2021. 1, 2, 4, 7