Towards Long-Form Video Understanding

Chao-Yuan Wu Philipp Krähenbühl

The University of Texas at Austin

Abstract

Our world offers a never-ending stream of visual stimuli, yet today's vision systems only accurately recognize patterns within a few seconds. These systems understand the present, but fail to contextualize it in past or future events. In this paper, we study long-form *video understanding. We introduce a framework for modeling long-form videos and develop evaluation protocols on large-scale datasets. We show that existing state-of-the-art short-term models are limited for long-form tasks. A novel object-centric transformer-based video recognition architecture performs significantly better on* **7** *diverse tasks. It also outperforms comparable state-of-the-art on the AVA dataset.*

1. Introduction

Our world tells an endless story of people, objects, and their interactions, each person with its own goals, desires, and intentions. Video recognition aims to understand this story from a stream of moving pictures. Yet, top-performing recognition models focus exclusively on short video clips, and learn primarily about the *present* — objects, places, shapes, *etc*. They fail to capture how this present connects to the *past* or *future*, and only snapshot a very limited version of our world's story. They reason about the '*what*', '*who*', and '*where*' but struggle to connect these elements to form a full picture. The reasons for this are two fold: First, short-term models derived from powerful image-based architectures benefit from years of progress in static image recognition [8, 72]. Second, many current video recognition tasks require little long-term temporal reasoning [36, 39, 61].

In this paper, we take a step towards leveling the playing field between short-term and long-term models, and study *long-form video understanding* problems (Fig. 1). First, we design a novel object-centric long-term video recognition model. Our model takes full advantage of current image-based recognition architectures to detect and track all objects, including people, throughout a video, but additionally captures the complex synergies among objects across time in a transformer-based architecture [74], called *Object Transformers*. Tracked instances of arbitrary length along with their visual features form basic semantic elements. A transformer architecture then models arbitrary interactions between these elements. This object-centric design takes inspiration from early work that builds space-time instance representations [5, 14, 22, 87], but further considers more complex inter-instance interactions over a long span of time. The model can be trained directly for a specific end-task or pre-trained in a self-supervised fashion similar to models in image recognition [10, 26, 53, 55, 70] and language understanding [13, 40, 45, 86].

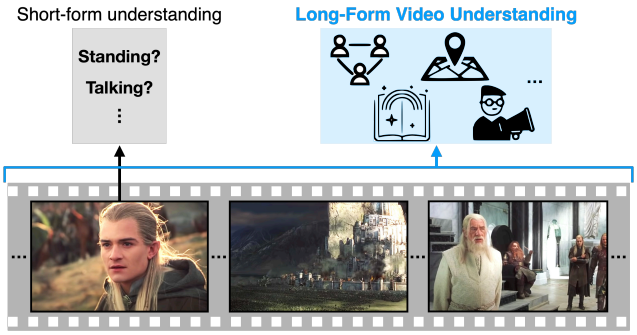


Figure 1. **Long-Form Video Understanding** aims at understanding the "full picture" of a long-form video. Examples include understanding the storyline of a movie, the relationships among the characters, the message conveyed by their creators, the aesthetic styles, *etc*. It is in contrast to 'short-form video understanding', which models short-term patterns to infer local properties.

Second, we introduce a large-scale benchmark, which comprises of 9 diverse tasks on more than 1,000 hours of video. Tasks range from content analysis to predicting user engagement and higher-level movie metadata. On these long-form tasks, current short-term approaches fail to perform well, even with strong (Kinetics-600 [6], AVA [25]) pre-training and various aggregation methods.

Our experiments show that *Object Transformers* outperform existing state-of-the-art methods on most of the long-form tasks, and significantly outperform the current state-of-the-art on existing datasets, such as AVA 2.2. The videos we use are publicly available and free.

2. Related Work

Short-form video understanding has seen tremendous progress in both efficiency [15, 73, 82, 92] and accuracy [8, 16, 60, 77] in recent years. Most state-of-the-art models are based on 2D or 3D CNNs operating on short videos of less than five seconds [8, 15, 16, 60, 73, 77, 82, 91, 92]. A few works explore long-term patterns for improving local pattern recognition [58, 81], but not long-form understanding.

Long-form video understanding is less explored. It aims to understand the full picture of a much longer video (*e.g.*, minutes or longer). Tapaswi *et al*. [68] introduce a movie question answering dataset based on both text and video data. The benchmark, however, is dominated by language-only approaches [68], making it less ideal for evaluating progress of computer vision. Vicol *et al*. [75], Xiong *et al*. [84], and Huang *et al*. [30] use vision-only movie understanding datasets, but their videos are not publicly accessible due to copyright issues. Bain *et al*. [3] and Zellers *et al*. [88] propose joint vision-language benchmarks for text-to-video retrieval and question answering, respectively.

In this paper, we introduce a new long-form video understanding benchmark of 9 vision-only tasks on more than 30K freely accessible videos. Our evaluation is relatively simple compared to prior work that involves language components in evaluation protocols.

Some studies propose efficient architectures [31, 35, 91] or pooling-based methods [17, 19, 76] that may operate on long-form videos. These methods primarily focus on the interactions between adjacent frames, while our model captures the long-range interactions between tracked objects.

Representing instances as space-time trajectory has a long history in computer vision [14, 21, 22, 56]. Our work takes inspiration from these concepts, but further considers inter-instance relationships in our methods.

Interaction modeling for images is widely studied for improving, *e.g.*, object detection [11], human action recognition [20], or 3D recognition [89]. For videos, a growing line of work models interactions among objects or features for improving short-term recognition [4, 18, 33, 47, 48, 50, 69, 78, 90]. They mainly leverage spatial but not temporal structures of a video.

Self-supervised learning drives the success of natural language processing models [13, 40, 45], visual pattern learning [10, 23, 26, 32, 49, 53, 55, 79], and image-language joint representation learning [12, 42, 46, 62, 66]. Some of these methods are video-based like ours, but aim at learning robust *spatial* rather than temporal features [23, 32, 49, 79]. For example, Jabri *et al*. [32] track spatial features across frames to learn viewpoint-, scale-, or occlusion-invariant features for each instance. Instead, our goal is to learn long-term and high-level interaction patterns. Several other papers leverage multiple modalities for learning joint concepts [2, 38, 51, 62, 63]. Our method requires only visual data. Sun *et al*. [64] recently propose a joint language-vision model for learning long-term concepts on cooking videos. It shares a similar goal to our approach. The main difference is that they use a 'frame-as-word', 'video-as-sentence' analogy, while we build object-centric representations. Our model captures interactions between objects, while a 'frame-as-word' approach captures the interactions between adjacent video frames. We will show the significance of this design in experiments.

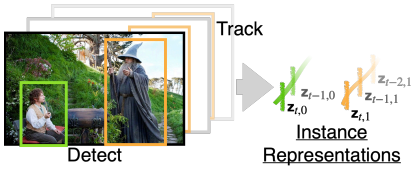


Figure 2. We leverage short-term detection and tracking to form instance representations.

3. Preliminaries

Existing short-term models parse many aspects of a video. They detect objects, track boxes, *etc*. This local understanding forms a useful building block for our Object Transformers. Instead of "re-learning" these short-term concept from scratch, our method builds on these short-term recognition modules. We briefly review these methods and introduce notations below.

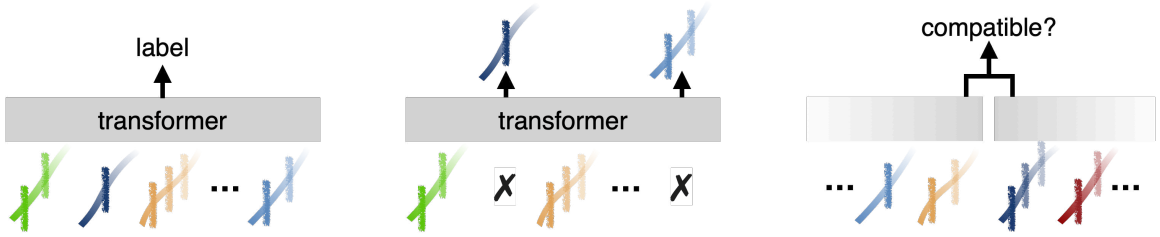
Action and Object Detection. The states and properties of humans and objects take a central role in the story told by the visual world. In this paper, we use an action detection model [16] to recognize the atomic actions [25] of humans, and an object detector [57] to find objects with their categories. We denote the bounding box of a detected person or object i at frame t by $\mathbf{s}_{t,i} \in \mathbb{R}^4$, and the associated feature representation by $\mathbf{z}_{t,i}$.

Tracking. An instance often appear in multiple frames. Tracking algorithms track these appearances over time and associate them to their identity [52, 71]. We use $\tau_{t,i}$ to denote the associated instance index of detection i at time t.[1]

Shot Transition Detection. Shot transitions or "cuts" segment a video into shots. They form natural semantic boundaries. A rule-based thresholding strategy typically suffices for shot transition detection [9]. c_u denotes the shot an instance u is in.

The methods above parse local properties of a video but do not connect them to form a more complete picture of the whole video. We tackle this issue next.

[1] In instance segmentation literature [27], the term 'instance' often refers to one appearance in one frame. We extend the definition and use 'instance' to refer to one appearance in a space-time region, which may comprise multiple frames.



(a) **Long-Form Understanding** (b) **Masked-Instance Pre-Training** (c) **Compatibility Pre-Training**

Figure 3. **Long-Form Video Understanding with Object Transformers.** Object Transformers take in instance-level representations and model the synergy among them for long-form video tasks (3a). To address the sparsity in supervising signals, we pre-train the model to predict the semantic representations of randomly masked instances (3b) and/or predict "compatibility" between two videos (3c). Both pre-training tasks encourage Object Transformers to learn long-term semantics, commonsense, or human social behaviors.

4. Long-Form Video Understanding

We propose *Object Transformers* for long-form video understanding. It builds on two key ideas: 1) An object-centric design, and 2) a self-supervised learning approach.

4.1. Object-Centric Design

Instead of modeling videos as width×height×time volume of pixels, we take a more structured approach and model *how each instance evolves in space and time* and the *synergy between the instances*.

Consider a set of instances $\mathcal{U}$ (people, tables, cars, ...) found and tracked by short-term models (§3). Each instance $u \in \mathcal{U}$ is associated with features in space-time $\{(t, \mathbf{s}_{t,i}, \mathbf{z}_{t,i}) \mid \tau_{t,i} = u, \forall t, i\}$ (Fig. 2), where t, $\mathbf{s}_{t,i}$, $\mathbf{z}_{t,i}$, and $\tau_{t,i}$ denote the time stamp, spatial locations, short-term features, and the tracked identity, respectively.

We build a transformer-based architecture [74] to model both how each instance $u \in \mathcal{U}$ evolves and interacts with other instances. The transformer takes a set of representation vectors as input. In our case, each vector correspond to a box-level representation together with its position, link and shot information. Namely, for each $(t', \mathbf{s}', \mathbf{z}')$ associated with u, we construct one input vector

$$\begin{aligned}\mathbf{y}' :=& \mathbf{W}^{(\text{feat})}\mathbf{z}' + \mathbf{W}^{(\text{spatial})}\mathbf{s}' + \mathbf{E}^{(\text{temporal})}_{t'} \\ &+ \mathbf{E}^{(\text{instance})}_{u} + \mathbf{E}^{(\text{shot})}_{c_u} + \mathbf{b}, \end{aligned} \tag{1}$$

where the matrices $\mathbf{W}^{(\text{feat})}$ and $\mathbf{W}^{(\text{spatial})}$ project $\mathbf{z}'$ and $\mathbf{s}'$ into a shared 768-dimensional vector space, and $\mathbf{b}$ is a bias term. $\mathbf{E}^{(\text{temporal})}$ and $\mathbf{E}^{(\text{shot})}$ are position embeddings [13] indexed by 'time stamp' and 'shot index', respectively. We additionally add a learned instance-level embedding vector $\mathbf{E}^{(\text{instance})}$ so that the model knows what inputs belong to the same instance. However, learning instance-specific embeddings cannot generalize to new videos with unseen instances. We thus randomly assign instance indices at each forward pass. This encourages the model to leverage only "instance distinctiveness" rather than memorizing instance-specific information. The exact model specification is given in the Supplementary Material.

We use a learned vector $\mathbf{E}^{[\text{CLS}]}$ to be the first token of each example (similar to the "[CLS]" special token in Devlin *et al.* [13]), and use the output vector corresponding to that position, $\mathbf{v}^{[\text{CLS}]}$, as the video-level representation. We use a linear output head $h^{(\text{task})}(\mathbf{v}^{[\text{CLS}]})$ to perform each video-level end-task. Fig. 3a illustrates our model. In §4.2, we will introduce additional output heads, $h^{(\text{mask})}$ and $h^{(\text{compat})}$ along with the associated loss functions $\ell^{(\text{mask})}$ and $\ell^{(\text{compat})}$ for pre-training object transformers in a self-supervised manner.

Discussion: Object-Centric *vs*. Frame-Centric *vs*. Pixel-Volume. Most existing methods either view a video as a list of 2D images (*e.g.*, [35, 64]) or a width×height×time pixel volume (*e.g.*, [8, 16, 72]). While these views are convenient, we argue that they are unnatural ways to look at the signals, possibly leading to difficulties in learning. After all, a video frame is simply a projection of (constantly changing) objects and scenes in a 3D world snapshotted at a particular point of time. Modeling videos through modeling the interactions among a list of 2D images likely suffers from model misspecification, because the projected 2D images do not interact with each other — It is the objects in our 3D world that interact with each other. Object Transformers directly model these interactions.

Modeling a video as a width×height×time pixel volume [72] amplifies the problem even more, especially for long videos, because the pixel volume is simply a stack of the 2D projections, with arbitrary camera positions. The semantic of the viewed world is however, invariant to these artifacts introduced by observers. The pixel-volume view ignores this invariance, and thus likely hurts data efficiency, let alone the prohibitive cost to scale 3D CNNs to long-form videos. Object Transformers leverage tracking and avoid these issues. In §6, we will empirically demonstrate the advantage of the object-centric design over existing frame-list or pixel-volume designs.

4.2. Self-Supervision

Long-form video understanding also brings challenges in supervision. Intuitively, long-form videos could enjoy less 'supervising signal' per pixel, given its potentially larger number of pixels per annotation. In §6, we will see that a long-form video model trained from scratch indeed suffers generalization challenges in many tasks. One way to alleviate the supervision issue is to first pre-train our model in a self-supervised fashion on unlabeled videos, before fine-tuning on the end-tasks.[2] We present two pre-training strategies below.

1) Masked-Instance Prediction. One natural choice of the pretext task is a masked instance prediction task, similar to Context Encoders [55] or BERT [13]. Namely, we mask out features of randomly selected instances $\mathcal{M} \subset \mathcal{U}$, and train our Object Transformers to predict the "semantics" (*e.g.*, object categories, person actions) of the masked instances. Note that we mask out only the feature vector $\mathbf{z}$, but retain time stamp t, spatial position $\mathbf{s}$, instance embedding $\mathbf{E}^{(\text{instance})}$, and shot embedding $\mathbf{E}^{(\text{shot})}$ for specifying 'where in a video' to predict. Following standard practice [13], masked feature $\mathbf{z}$ is replaced by a learned embedding $\mathbf{z}^{(\text{mask})}$ with 80% probability, replaced by a randomly sampled feature with 10% probability, and stays unchanged with the remaining 10% probability. At each output position corresponding to the masked inputs, we use a output head $h^{(\text{mask})}$ to predict a probability vector $\hat{\mathbf{p}} \in \Delta^{d-1}$ that regresses the pseudo-label[3] $\mathbf{p} \in \Delta^{d-1}$ using distillation loss [29] (with temperature $T = 1$),

$$\ell^{(\text{mask})}\left(\mathbf{p}, \hat{\mathbf{p}}\right) := \sum_{k=0}^{d-1} -\mathbf{p}_k \log\left(\hat{\mathbf{p}}_k\right). \tag{2}$$

Fig. 3b presents a visual illustration. Intuitively, this task asks '*What object might it be?*' or '*What a person might be doing?*' in the masked regions, given the context. We humans can perform this task very well, given our social knowledge and commonsense. We train Object Transformers to do the same.

Discussion: Masked-Instance Prediction *vs*. Masked-Frame Prediction. Our method share similar spirits with Sun *et al*. [64]'s 'Masked-Frame Prediction' pretext task (if removing their language components), but with important distinctions. 'Masked-Frame Prediction' is in fact quite easy, as linear interpolation using the 2 adjacent frames already provides a strong solution in most cases due to continuity of physics. This observation is consistent with the observations in Sun *et al*. [64] that such pre-training method is data hungry and that their 'visual-only' variant (without using language) is not effective. Masked-Instance Prediction, on the other hand, does not suffer from the trivial solution. It directly models how objects interact with each other, rather than learning to interpolate in the projected 2D space.

Discussion: Masked-Instance Prediction *vs*. Spatial-Feature Learning Methods. Also note that our goal of pre-training is different from the goal of most prior work on self-supervised method on videos [2,23,32,38,49,79]. They typically involve tracking an object or an interest point over time to learn (*e.g.*, view-point, scale, occlusion, lighting) invariance on one instance. Their goal is learning robust *spatial* representations. In this paper, we aim at learning longer-term patterns in videos.

2) Span Compatibility Prediction. The second pre-training pretext task we use is to classify whether two spans of video are "*compatible*". For example, we may define two spans to be compatible when they come from the same scene or happen one after another. To solve this task, the model is encouraged to learn high-level semantics concepts, *e.g.*, '*wedding ceremony*' should be more compatible with '*party*' and '*dinner*' than '*camping*' or '*wrestling*'. Fig. 3c illustrates this method. We use an output head $h^{(\text{compat})}$ to obtain $\mathbf{v} = h^{(\text{compat})}\left(\mathbf{v}^{[\text{CLS}]}\right)$ and use the InfoNCE loss [54] for compatibility training:

$$\ell^{(\text{compat})}\left(\mathbf{v}, \mathbf{v}^+, \mathbf{v}^-\right) = -\log \frac{e^{\left(\mathbf{v}\cdot\mathbf{v}^+\right)}}{e^{\left(\mathbf{v}\cdot\mathbf{v}^+\right)} + \sum_{n=0}^{N-1} e^{\left(\mathbf{v}\cdot\mathbf{v}_n^-\right)}}, \tag{3}$$

where $\mathbf{v}^+$ and $\mathbf{v}_n^-$ correspond to the spans compatible and incompatible with $\mathbf{v}$, respectively.

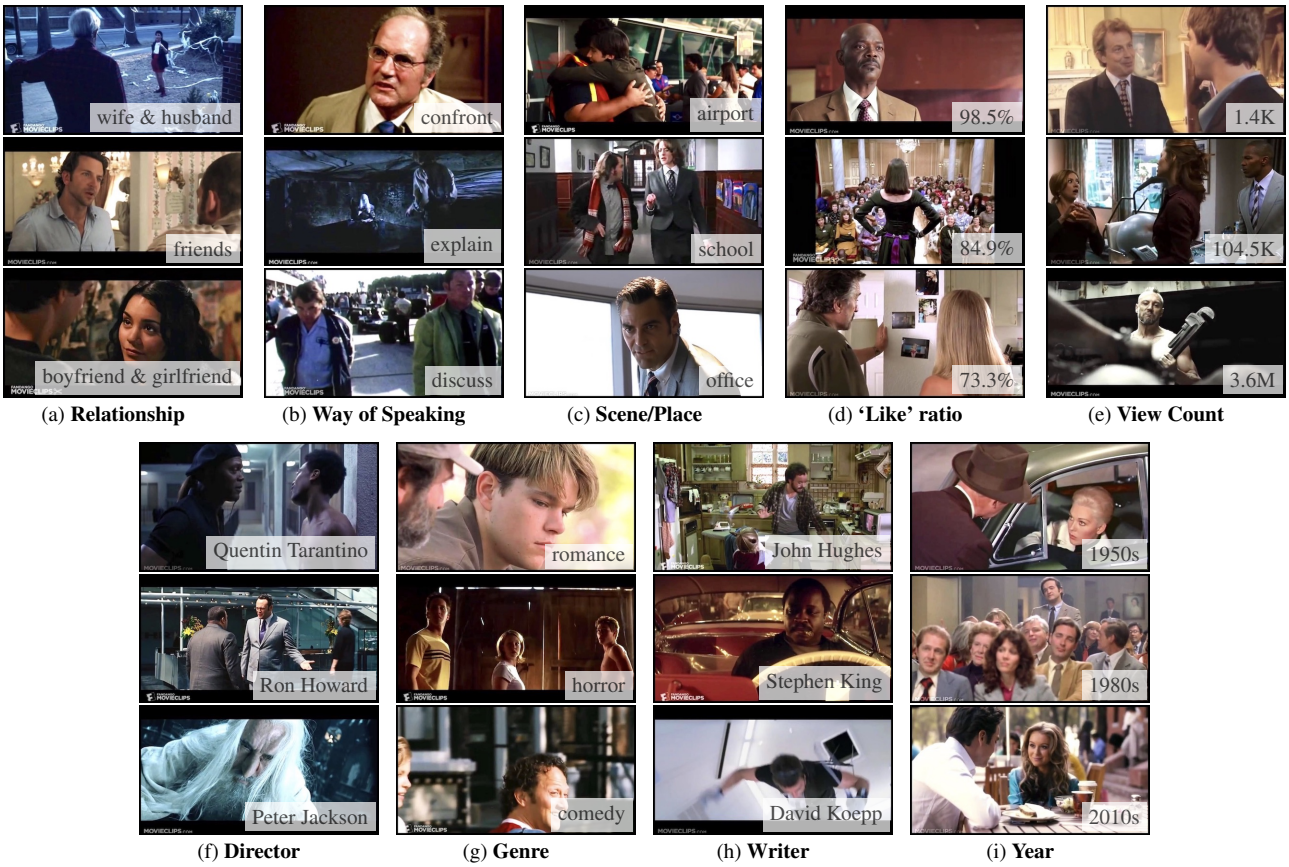
Discussion: Comparison to Next-Sentence Prediction. Compatibility prediction is a modified version of the "next-sentence prediction" task commonly used in NLP [13]. One distinction is that while languages typically have strict grammar and rich structures, videos are more flexible in structure. For example, an event of 'dinner' can take place with arbitrary number of people for arbitrarily long and potentially presented in multiple shots in a video. We thus relax the requirement of predicting immediate adjacency, and enforce a more relaxed "compatibility" objective. We will describe our exact instantiation in §4.3.

4.3. Implementation Details

Instance Representations. We use a Faster R-CNN [57] with ResNet-101 [28] backbone and FPN [43] pre-trained on COCO [44] to find objects other than humans. The model obtains 42.0 box AP on COCO. We use the RoIAlign [27] pooled feature vector and the end of the Faster-RCNN as feature vector $\mathbf{z}$. For person action detection, we adopt a Faster-R-CNN-based person detector [57]

[2]These pre-training methods are self-supervised, because they do not require additional annotations to perform. Our full approach is not self-supervised, because the short-term features are learned from labeled data.

[3]We infer pseudo-labels using the same short-term model that is used to compute the feature vector $\mathbf{z}$.



(a) **Relationship** (b) **Way of Speaking** (c) **Scene/Place** (d) **'Like' ratio** (e) **View Count**

(f) **Director** (g) **Genre** (h) **Writer** (i) **Year**

Figure 4. **The Long-Form Video Understanding (LVU) Benchmark.** Here we present three examples with their annotations for each task. LVU contains a wide range of tasks for probing different aspects of video understanding research and model design. The full list of classes for each task, more details, and more examples are available in Supplementary Material.

(~93.9 AP@50) commonly used in prior work [16, 81] to detect people first, and use a ResNet-101 [28] Slow-Fast network [16] with non-local blocks [77] to compute RoIAlign [27] pooled features as $\mathbf{z}$ for each person box. The model is pre-trained on AVA [25] and achieves 29.4% mAP on the AVA validation set. We represent $s_{i,j}$ as the positions of the four corners $\left(s_{i,j}^{(\text{top})}, s_{i,j}^{(\text{bottom})}, s_{i,j}^{(\text{left})}, s_{i,j}^{(\text{right})}\right)$, where each of the values are normalized in $[0, 1]$. For tracking, we adopt the algorithm described in Gu *et al*. [25]. We use PySceneDetect [9] for shot transition detection.

Compatibility Prediction. The MovieClips dataset [1] we use contain (typically one-to-three-minute-long) segments of movies. In this paper, we define two spans to be compatible if they come from the same segment.

When training with compatibility prediction, each mini-batch of size n comprises $n/2$ pairs of positive examples $(\mathbf{v}, \mathbf{v}^+)$. Each pair uses all other examples in the same mini-batch as negative examples $\mathbf{v}^-$.

Output Heads. Following prior work [13], $h^{(\text{mask})}$ is a 2-layer MLP. $h^{(\text{compat})}$ and all end tasks use dropout with rate 0.1 followed by a linear layer.

5. The Long-Form Video Understanding (LVU) Benchmark

We introduce a new benchmark that contains 9 tasks for evaluating long-form video understanding. The benchmark is constructed on the publicly available MovieClips dataset [1], which contains ~30K videos from ~3K movies.[4] We resize all videos such that their height is 480 pixels. Each video is typically one-to-three-minute long.

Tasks. Our tasks cover a wide range of aspects of long-form videos, including **content understanding** (*'relationship'*, *'speaking style'*, *'scene/place'*), **user engagement prediction** (*'YouTube like ratio'*, *'YouTube popularity'*), and **movie metadata prediction** (*'director'*, *'genre'*, *'writer'*, *'movie release year'*). Fig. 4 presents examples of each task. For content understanding tasks, we parse the description associated with each video and use the most common discovered categories (*e.g.*, 'friends', 'wife & husband', *etc.*) to form a task (*e.g.*, 'relationship' prediction).

[4] Videos are accessed on February 26, 2020. Animations are excluded. Outros are removed for all videos.

	average	content			user engagement		metadata			
	rank	relation (↑)	speak (↑)	scene (↑)	like (↓)	views (↓)	director (↑)	genre (↑)	writer (↑)	year (↑)
R101-SlowFast+NL [16, 28, 77]	2.44	52.4	35.8	54.7	0.386	3.77	44.9	53.0	36.3	**52.5**
VideoBERT [64]	2.22	52.8±1.0	37.9±0.9	54.9±1.0	0.320±0.016	4.46±0.07	47.3±1.7	51.9±0.6	**38.5**±1.1	36.1±1.4
Object Transformer	**1.33**	**53.1**±1.4	**39.4**±1.2	**56.9**±1.0	**0.230**±0.005	**3.55**±0.05	**51.2**±0.8	**54.6**±0.6	34.5±0.9	39.1±1.2

Table 1. **Comparison to Prior Work.** Our Object Transformer outperforms both baselines by a clear margin in terms of the overall ranking. The results support that modeling the synergy across people and objects is important for understanding a long-form video. Interestingly, short-term models suffice to work well for *year prediction*, which matches our expectation, since the year can often be recognized through solely the picture resolution/quality (Fig. 4i). We report the average over 5 runs with standard error for VideoBERT and Object Transformer.

We use YouTube statistics for user engagement prediction tasks. For metadata prediction tasks, we obtain the metadata from the corresponding IMDb entries[5]. Task construction details, statistics, and more examples are available in Supplementary Material.

Evaluation Protocol. Content understanding and metadata prediction tasks are single-label classification tasks, evaluated by top-1 classification accuracy. User engagement prediction tasks are single-valued regression tasks, evaluated by mean-squared-error (MSE). Compared to existing tasks [3, 88] on this dataset, the output space and evaluation protocol of LVU is relatively simple. We hope this choice makes result interpretation easier. Each task is split into 70% for training, 15% for validation, and 15% for testing. Since we predict "movie" specific metadata for metadata prediction tasks, we make sure the three splits contain mutually exclusive sets of movies. We select hyperparameters based on validation results, and report all results on test sets.

6. Experiments

Pre-Training Details. We pre-train our models on the MovieClip videos for 308,000 iterations with a batch size of 16 (2 epochs of all possible, overlapping spans) using Adam [37], with a weight decay of 0.01 and a base learning rate of 10^{-4}. We use linear learning rate decay and linear warm-up [24, 28] for the first 10% of the schedule, following prior work [45]. We sample 60-second video spans for training our models.[6] Since each example contains a different number of instances of different lengths, we perform attention masking as typically implemented in standard frameworks [80].

End-Task Fine-Tuning Details. Following prior work [45], we perform grid search on training epochs and batch size $\in \{16, 32\}$ on validation sets. Detailed training schedule selected for each task is in Supplementary Material. We report the average performance over 5 runs in §6.1. We use a base learning rate of 2e-5 (the same as what is used in BERT [13]), which we find to work well for all tasks. Other hyperparameters are the same as pre-training.

[5] https://www.imdb.com/

[6] In preliminary experiments, we do not see advantages with a longer training schedule or using spans longer than 60 seconds.

6.1. Main Results

We start with evaluating different state-of-the-art existing methods on long-form tasks, and comparing them with the proposed Object Transformers.

Compared Methods. The most prominent class of video understanding methods today is probably 3D CNNs with late fusion [8, 16, 72, 73, 77, 82], which has been widely used for a wide range of tasks [36, 39, 59, 61]. To compare with this category of methods, we use a large state-of-the-art model, a ResNet-101 [28] SlowFast network [16] with non-local blocks [77] running on 128 frames, pre-trained on Kinetics-600 [6] and AVA [25] as a baseline method. We train the network using SGD with cosine learning rate schedule, linear warmup [24], and a weight decay of 10^{-4}, following standard practice [16]. We select the base learning rate and the number of training epochs on validation set for each task; More details are in Supplementary Material.

Another promising baseline we compare to is the recently proposed frame-based long-term models, VideoBERT [64]. We compare with its vision-only variant, since language is beyond the scope of this paper.[7]

Results. Tab. 1 shows that our Object Transformer outperforms both baselines by a clear margin in terms of the overall ranking. The short-term model ('R101-SlowFast+NL'), is not able to perform well even with a large backbone and strong pre-training (Kinetics-600 [6] and AVA [25]). This validates the importance of long-term modeling. We also observe that object-centric modeling (Object Transformers) is advantageous compared with frame-centric modeling ('VideoBERT' [64]). Interestingly, short-term models suffice to work well for *year prediction*. This should not come as a surprise, since local statistics such as image quality or color style already capture a lot about the '*year*' of a video (*e.g.*, as shown in Fig. 4i). VideoBERT [64] works well for *writer* prediction, suggesting that this task might not require too much detailed interaction modeling.

In short, a long-term and object-centric design is important for a wide range of LVU tasks.

[7] We reached out to the authors of VideoBERT [64], but they were not able to share the code with us. We thus present results based on our re-implementation. We select hyperparameters for VideoBERT [64] with the same grid-search protocol as our method for fair comparison. More implementation details are in Supplementary Material.

pre-train	relation	speak	scene	like↓	views↓	director	genre	writer	year
None	46.9	39.8	53.8	0.262	3.44	43.0	55.8	34.5	35.0
Mask	**54.7**	**40.3**	58.0	0.238	3.71	53.3	**56.1**	**35.1**	**40.6**
Mask+Compat	50.0	32.8	**60.0**	**0.234**	**3.37**	**58.9**	49.3	32.7	39.9
Δ	(+7.8)	(+0.5)	(+6.2)	(-.028)	(-.07)	(+15.9)	(+0.3)	(+0.6)	(+5.6)

(a) **Pre-training**

	relation	speak	scene	like↓	views↓	director	genre	writer	year
Short-term	50.0	40.3	52.9	0.366	3.57	54.2	52.9	28.6	37.8
Avg pool	37.5	36.8	57.1	0.496	3.82	40.2	54.4	**37.5**	32.9
Max pool	50.0	37.8	58.8	0.284	3.78	52.3	55.8	32.7	34.3
Transformer	**54.7**	**40.3**	**60.0**	**0.234**	**3.37**	**58.9**	**56.1**	35.1	**40.6**

(b) **Long-term module**

	relation	speak	scene	like↓	views↓	director	genre	writer	year
Person	54.7	**40.3**	**60.0**	0.234	**3.37**	**58.9**	**56.1**	35.1	40.6
Person+Obj.	54.7	37.8	58.8	**0.223**	3.67	48.6	55.8	**36.3**	**42.0**

(c) **Modality**

	relation	speak	scene	like↓	views↓	director	genre	writer	year
10k	50.0	**40.8**	58.0	**0.230**	3.42	53.3	53.2	32.7	37.8
30k (all)	**54.7**	40.3	**60.0**	0.234	**3.37**	**58.9**	**56.1**	**35.1**	**40.6**

(d) **Number of pre-training videos**

Table 2. **Ablation Experiments.** Our results validate that self-supervised pre-training brings consistent gains across tasks (2a). We also observe that simpler pooling methods are not as effective as transformer, supporting that object-level interaction modeling is beneficial (2b). Modeling non-person objects is beneficial for a few tasks, but modeling humans along is already strong is most of the tasks (2c). Finally, pre-training on more data helps in most cases (2d), suggesting promising future work using even larger datasets. (↓: lower is better)

6.2. Ablation Experiments

Pre-Training. We first evaluate the impact of the proposed pre-training methods. Tab. 2a shows that on all tasks we evaluate, pre-training is beneficial.[8] In particular, Masked Instance Pre-Training alone works well in almost all tasks, while adding Compatibility Pre-Training helps in 4 out of the 9 tasks. Interestingly, our results are similar to observations in NLP research, where the 'masked-language model' alone works well on some tasks (*e.g.*, [34]), while additionally using 'next-sentence-prediction' helps on others (*e.g.*, [13]). In other parts of this paper, we use the best performing pre-training method (selected based on validation results) as the default for each task.

Long-Term Module. Most existing methods perform either pooling-based aggregation [76] or no aggregation at all (late fusion only) [16, 77] when it comes to long-term modeling. Tab. 2b compares our Object Transformer with these approaches. All methods in Tab. 2b build on the same input features. The only difference is the module built on top of these features. Object Transformer works better on 8 out of the 9 tasks, showing that for long-form video understanding, a more powerful object-level interaction modeling is advantageous. Interestingly, for '*movie writer*' prediction, a transformer does not outperform even average pooling. We conjecture that writer prediction might require a higher level of cognition or abstraction ability, that is beyond what transformers can do. We think studying this task is interesting future work.

Modality. While humans are arguably the most central elements for understanding a video, we study the benefit of including other objects. Tab. 2c shows that adding objects brings only mild improvement on three tasks. This suggests that human behavior understanding plays an crucial role in most long-form tasks.

[8] In ablation experiments, we report the results without averaging over 5 runs due to computation resource constraints. Thus the results are slightly different from the results reported in Tab. 1.

	Action Detection	Object Detection	Transformer
params (M)	59.2	60.6	27.0
FLOPs (G)	242.0	88.6	1.8

Table 3. **Inference Complexity Breakdown.** Object Transformer is small and efficient — taking only 0.7% of the FLOPs and being 2.2× smaller compared to Action Detection.

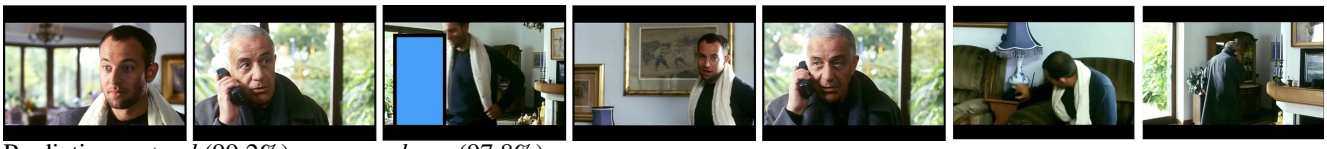
Number of Pre-Training Videos. A great advantage of our pre-training methods is that they are self-supervised, not requiring any human annotations. It is thus relatively easy to pre-train on large-scale datasets. Tab. 2d shows that on most tasks, pre-training on more data helps, suggesting promising future research that leverages even more data.

Model Complexity. Tab. 3 presents a breakdown analysis for complexity in terms of both model size and FLOPs. We see that our Object Transformer is small and efficient. It is 2.2× smaller and takes only 0.7% of the FLOPs compared to short-term action detection. We thus expect future research on long-form video understanding to be accessible.

Example Predictions of Masked Instance Prediction. Finally, we present case study to understand what the model learns with Masked Instance Prediction. Fig. 5 presents three examples from a hold-out dataset of AVA [25] along with the model outputs. We see that our model leverages the context, some of them not on the same frame, and make sensible predictions without seeing the actual content. This validates that long-term human interactions can indeed be learned in a self-supervised way.

6.3. Experiments on AVA

So far we have evaluated Object Transformers on long-form video tasks. We next evaluate its ability of improving "short-form" recognitions through incorporating long-term context. Here evaluate our method on AVA [25] V2.2 for spatial-temporal action detection. Performance is measured by mAP, following standard practice [25].



Predictions: *stand* (99.2%), *answer phone* (97.8%)

Predictions: *hand clap* (79.1%), *stand* (71.5%), *watch (a person)* (69.4%)

Predictions: *play musical instrument* (90.1%), *sit* (56.9%)

Figure 5. **Masked Instance Prediction Examples.** Here we present three examples along with their masked instances, and the actions of these instances predicted by our model.† Without seeing the actual content, our model leverages long-term context and makes plausible predictions. Some of these (*e.g.*, the top example) are not possible without modeling longer-term context. (Best viewed on screen.)
†: Here we list predictions with ≥50% probabilities. We present only 7 frames for each example at 0.5 FPS due to the space constraint; See Supplementary Material for the full examples.

	input	pre-train	mAP	FLOPs
AVA V2.1				
AVA [25]	V+F	K400	15.6	-
ACRN [65]	V+F	K400	17.4	-
STEP [85]	V	K400	18.6	-
Zhang *et al.* [90]	V	K400	22.2	-
RTPR [41]	V+F	ImageNet	22.3	-
Girdhar *et al.* [18]	V	K400	25.0	-
LFB [81]	V	K400	27.7	-
AVSlowFast [83]	V+A	K400	27.8	-
AVA V2.2				
AVSlowFast [83]	V+A	K400	28.6	-
AIA [67]	V	K700	32.3	-
X3D-XL [15]	V	K600	27.4	-
SlowFast R101 [16]	V	K600	29.4	1.000×
Object Transformer (masked)	V	K600	29.3	1.007×
Object Transformer	V	K600	**31.0**	1.007×

Table 4. **Action Recognition Results on AVA.** Object Transformer outperforms prior work, which use only short-term information. Our results suggest that long-term interaction and context are beneficial for short-form tasks as well. (V: Visual; A: Audio; F: Flow; K400: [36]; K600: [6]; K700: [7].)

Adaptation to AVA. Note that directly fine-tuning Object Transformers to predict the box-level AVA outputs would lead to a "short-cut" solution, where the model directly looks at the corresponding input feature $\mathbf{z}$ without long-term modeling. We thus mask out the input features $\mathbf{z}$ for the target instances (similar to masked-instance pre-training) when fine-tuning the AVA model. This, however, would put our model at a disadvantage, since prediction given only context is much harder than the original task. We thus take a simple approach of late fusing the short-term prediction, and fine-tuning only the final linear layer (for 2000 iterations using a base learning rate of 10^{-4}). This procedure is efficient, as no updating of the attention layers are involved.

Results. We evaluate and compare our Object Transformer with prior work in Tab. 4. We see that without using optical-flow [25, 41, 65], audio [83], or task-specific engineering, our Object Transformer outperforms state-of-the-art short-term models that use comparable (K600 [6]) feature pre-training by **1.6%** absolute (29.4% → **31.0**%). This shows that even for short-form tasks, it is beneficial to consider long-term context to supplement or disambiguate cases where local patterns are insufficient. Note that this improvement comes almost "for free", as it only uses **0.7%** of additional FLOPs and fine-tuning of a linear layer. Interestingly, a "masked" Object Transformer *without* late fusion (denoted 'Object Transformers (masked)' in Tab. 4), still achieves 29.3, demonstrating that "*given context only*", our model is able to leverage context and predict the semantics of the masked parts of a video with state-of-the-art quality (also see Fig. 5 for qualitative results).

In short, plugging a Object Transformer into a short-form task is easy, and it leads to a clear improvement.

7. Conclusion

In this paper, we take a step towards understanding long-form videos. We build a new benchmark with 9 tasks on publicly available large datasets to evaluate a wide range of aspects of the problem. We observe that existing short-term models or frame-based long-term models are limited in most of these tasks. The proposed Object Transformers that model the synergy among people and objects work significantly better. We hope this is a step towards deeper, more detailed, and more insightful understanding of our endlessly evolving visual world, with computer vision.

Acknowledgments. This material is based upon work supported by the National Science Foundation under Grant No. IIS-1845485 and IIS-2006820. Chao-Yuan was supported by the Facebook PhD Fellowship.

References

[1] MovieClips. https://www.movieclips.com/. Accessed: 2020-02-26.

[2] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *ICCV*, 2017.

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

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

[5] Aaron F. Bobick and James W. Davis. The recognition of human movement using temporal templates. *PAMI*, 2001.

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

[7] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019.

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

[9] Brandon Castellano. PySceneDetect, 2018.

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

[11] Xinlei Chen and Abhinav Gupta. Spatial memory for context reasoning in object detection. In *ICCV*, 2017.

[12] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: UNiversal image-TExt representation learning. In *ECCV*, 2020.

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

[14] Alexei A. Efros, Alexander C. Berg, Greg Mori, and Jitendra Malik. Recognizing action at a distance. In *ICCV*, 2003.

[15] Christoph Feichtenhofer. X3D: Expanding architectures for efficient video recognition. In *CVPR*, 2020.

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

[17] Basura Fernando, Efstratios Gavves, José Oramas, Amir Ghodrati, and Tinne Tuytelaars. Rank pooling for action recognition. *PAMI*, 2016.

[18] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *CVPR*, 2019.

[19] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. ActionVLAD: Learning spatio-temporal aggregation for action classification. In *CVPR*, 2017.

[20] Georgia Gkioxari, Ross Girshick, Piotr Dollár, and Kaiming He. Detecting and recognizing human-object interactions. In *CVPR*, 2018.

[21] Georgia Gkioxari and Jitendra Malik. Finding action tubes. In *CVPR*, 2015.

[22] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. *PAMI*, 2007.

[23] Ross Goroshin, Joan Bruna, Jonathan Tompson, David Eigen, and Yann LeCun. Unsupervised learning of spatiotemporally coherent metrics. In *ICCV*, 2015.

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

[25] 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.

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

[27] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

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

[29] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

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

[31] Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Timeception for complex action recognition. In *CVPR*, 2019.

[32] Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. In *NeurIPS*, 2020.

[33] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *CVPR*, 2020.

[34] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving pre-training by representing and predicting spans. *ACL*, 2020.

[35] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[36] 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.

[37] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[38] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018.

[39] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011.

[40] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. In *ICLR*, 2020.

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

[42] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, Daxin Jiang, and Ming Zhou. Unicoder-VL: A universal encoder for vision and language by cross-modal pre-training. In *AAAI*, 2020.

[43] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[44] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[45] 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.

[46] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019.

[47] 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.

[48] Effrosyni Mavroudi, Benjamín Béjar Haro, and René Vidal. Representation learning on visual-symbolic graphs for video understanding. In *ECCV*, 2020.

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

[50] Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. EGO-TOPO: Environment affordances from egocentric video. In *CVPR*, 2020.

[51] Arsha Nagrani, Chen Sun, David Ross, Rahul Sukthankar, Cordelia Schmid, and Andrew Zisserman. Speech2Action: Cross-modal supervision for action recognition. In *CVPR*, 2020.

[52] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016.

[53] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

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

[55] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[56] Alessandro Prest, Vittorio Ferrari, and Cordelia Schmid. Explicit modeling of human-object interactions in realistic videos. *PAMI*, 2012.

[57] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[58] Mykhailo Shvets, Wei Liu, and Alexander C Berg. Leveraging long-range temporal relationships between proposals for video object detection. In *ICCV*, 2019.

[59] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016.

[60] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014.

[61] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[62] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: Pre-training of generic visual-linguistic representations. In *ICLR*, 2020.

[63] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer. *arXiv preprint arXiv:1906.05743*, 2019.

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

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

[66] Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019.

[67] Jiajun Tang, Jin Xia, Xinzhi Mu, Bo Pang, and Cewu Lu. Asynchronous interaction aggregation for action detection. In *ECCV*, 2020.

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

[69] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+O: Unified egocentric recognition of 3d hand-object poses and interactions. In *CVPR*, 2019.

[70] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ICML*, 2020.

[71] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. 1991.

[72] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*, 2015.

[73] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *ICCV*, 2019.

[74] 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.

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

[76] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[77] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

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

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

[80] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. HuggingFace's transformers: State-of-the-art natural language processing. *ArXiv preprint arXiv:1910.03771*, 2019.

[81] 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.

[82] Chao-Yuan Wu, Ross Girshick, Kaiming He, Christoph Feichtenhofer, and Philipp Krahenbuhl. A multigrid method for efficiently training video models. In *CVPR*, 2020.

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

[84] Yu Xiong, Qingqiu Huang, Lingfeng Guo, Hang Zhou, Bolei Zhou, and Dahua Lin. A graph-based framework to bridge movies and synopses. In *ICCV*, 2019.

[85] Xitong Yang, Xiaodong Yang, Ming-Yu Liu, Fanyi Xiao, Larry S Davis, and Jan Kautz. STEP: Spatio-temporal progressive learning for video action detection. In *CVPR*, 2019.

[86] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. XLNet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, 2019.

[87] Alper Yilmaz and Mubarak Shah. Actions sketch: A novel action representation. In *CVPR*, 2005.

[88] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *CVPR*, 2019.

[89] Jason Y. Zhang, Sam Pepose, Hanbyul Joo, Deva Ramanan, Jitendra Malik, and Angjoo Kanazawa. Perceiving 3D human-object spatial arrangements from a single image in the wild. In *ECCV*, 2020.

[90] Yubo Zhang, Pavel Tokmakov, Martial Hebert, and Cordelia Schmid. A structured model for action detection. In *CVPR*, 2019.

[91] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018.

[92] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. ECO: Efficient convolutional network for online video understanding. In *ECCV*, 2018.