Look for the Change: Learning Object States and State-Modifying Actions from Untrimmed Web Videos

Tomáš Souček[1] Jean-Baptiste Alayrac[2] Antoine Miech[2]
Ivan Laptev[3] Josef Sivic[1]
[1]CIIRC CTU [2]DeepMind [3]ENS/Inria
tomas.soucek@cvut.cz

https://data.ciirc.cvut.cz/public/projects/2022LookForTheChange/

Abstract

Human actions often induce changes of object states such as "cutting an apple", "cleaning shoes" or "pouring coffee". In this paper, we seek to temporally localize object states (e.g. "empty" and "full" cup) together with the corresponding state-modifying actions ("pouring coffee") in long uncurated videos with minimal supervision. The contributions of this work are threefold. First, we develop a self-supervised model for jointly learning state-modifying actions together with the corresponding object states from an uncurated set of videos from the Internet. The model is self-supervised by the causal ordering signal, i.e. initial object state → manipulating action → end state. Second, to cope with noisy uncurated training data, our model incorporates a noise adaptive weighting module supervised by a small number of annotated still images, that allows to efficiently filter out irrelevant videos during training. Third, we collect a new dataset with more than 2600 hours of video and 34 thousand changes of object states, and manually annotate a part of this data to validate our approach. Our results demonstrate substantial improvements over prior work in both action and object state-recognition in video.

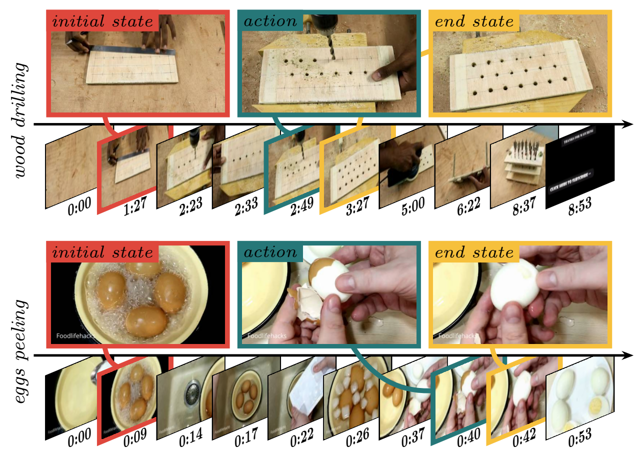


Figure 1. **Examples of object states and state-modifying actions** learnt by our model from a dataset of long uncurated web videos. In each example the top row shows: the initial state in the video (left), the state-modifying action (middle), and the end-state (right). The bottom row shows video frames sampled from the entire video with their corresponding timestamps. It illustrates the difficulty of finding the correct temporal localization of the object states and the actions in the entire video.

1. Introduction

Human actions often induce changes of the state of an object, as illustrated in Figure 1. Examples include "cutting an apple", "cleaning shoes", "tying a tie" or "filling-up a cup with coffee". People can easily recognize such actions and the resulting changes of object states [12], for example, when watching instructional videos. Furthermore, people can reproduce the actions in their environment, *e.g.* when following a recipe from a cooking video. However, artificial system with similar cognitive abilities is yet to be developed. Existing methods for recognizing object states and state-modifying actions address small-scale setups (5 objects and short manually curated videos) [3] or controlled environments [18]. At the same time, progress on automatic understanding of causal relations between actions and object states in the wild would be a major step in embodied video understanding and robotics. However, the task is challenging given the large amount and variability of existing object-action pairs as well as the difficulty of manually collecting and annotating video data for it.

In this paper, we investigate whether the learning of object states and corresponding state-modifying actions can be scaled-up to noisy uncurated videos from the web while

[1]Czech Institute of Informatics, Robotics and Cybernetics at the Czech Technical University in Prague.
[3]Département d'informatique de l'ENS, École normale supérieure, CNRS, PSL Research University, 75005 Paris, France.

using only minimal supervision. The contribution of this work is threefold as we outline below.

First, we develop a self-supervised model for jointly learning state-modifying actions and object states from an uncurated set of videos obtained from a video search engine. We explore the causal ordering in the video as a free supervisory signal and use it to discover the changing states of objects and state-modifying actions. We define it by the sequence of initial object state → manipulating action → end state, as illustrated in Figure 1. While the prior work on this problem [3] was limited to closed-form linear classifiers, our model is amenable to large-scale learning using stochastic gradient descent and supports non-linear multi-layer models.

Second, to cope with noisy uncurated data that may include a large proportion of irrelevant videos (*e.g*. videos of Apple laptops when learning "cutting an apple"), our model incorporates a noise adaptive weighting module that allows to filter out irrelevant videos. This noise adaptive weighting module is supervised by a small number of still images depicting the two states of the object, which are easy to collect using currently available image search engines. This attention mechanism allows us to scale our method to noisy uncurated data, as we show by experimental results.

Third, we collect a new "**ChangeIt**" dataset with more than 2600 hours of video and 34 thousand changes of object states. We manually annotate a portion of this data for evaluation. To validate our approach, we show results on this new uncurated dataset as well as on the existing smaller curated video dataset from [3]. We ablate key components of our method and demonstrate substantial improvements over prior work both in action and object state localization. The dataset, the code, and a trained model are publicly available.

2. Related work

Video and Language. A large body of work in automatic video understanding studies the use of natural language or speech data to train models for action and object state recognition. Prior work [4, 16, 19, 28, 29, 43, 45, 50, 53, 60, 64, 67] leveraged image and video description datasets [37, 45, 51, 55, 65, 71] to learn a joint vision-language embedding space, where visual and textual data are semantically aligned. In particular, [43, 53, 64] observed that object state and action recognition implicitly emerges, to some extent, from these vision and language models. In fact, the aligned vision and text training data often provides detailed descriptions of actions, objects and their different states. In contrast to these works, we explicitly model the causal nature of actions and their impact on object states in order to leverage this strong inductive bias in our model.

Object attributes and action modifiers. Learning object attributes (*e.g*. *sliced*, *diced*) has been approached in a supervised manner in still images [46–48, 52, 69] with the focus on the compositional nature of the attributes. Similarly, others have studied learning modifiers of actions (*e.g*. *quickly*) [21] from short clips (20 seconds) mined from web-instructional videos. Related to this, Doughty *et al*. [20] analyzed how the visual changes of object states can be used for skill determination in videos. The compositionality of natural language has also been explored for learning factored video-language embeddings for actions, objects and their attributes for retrieval applications [64] or to learn a contextualized language-object embedding [8]. Explicit models of changes in object states and the associated state-modifying actions have been explored in egocentric videos [24, 38]. Others have considered significantly reducing the amount of supervision by learning object states from web images gathered by querying a web-image search engine [32]. Closely related to us, there is the work of [22] that used a temporal cycle consistency loss between text and vision in instructional videos to find better targets for next frame prediction. By doing so, they implicitly discover potential object state changes but do not quantitatively evaluate the correctness and quality of those. Others directly focus on unsupervised learning of object states and the state-modifying actions from video [3, 18]. However, their work covers only a small-scale learning from a set of trimmed and curated videos [3] or a constrained scenario of videos observing a single specific scene [18]. In contrast, we consider large-scale learning from noisy untrimmed videos from the web.

Ordering as a form of supervision. The arrow of time is a strong signal [63] to learn about actions. Indeed, many actions happen in a certain order [7]. For example, you need toopen a bottle before being able to pour something from it. This can be used as a source of supervision. Past work [2, 10, 11, 35, 56, 62, 70] has leveraged such supervision to discover and temporally localize actions in untrimmed videos. Similarly, the natural ordering of recurring events has been used to distinguish key events from the background [74]. [72] trained a generative model from time-lapse videos to generate the future state of an object altered by time. Others have looked at the related task of next frame or action prediction as another form of supervision [15, 22, 27, 39, 40, 44, 54, 58]. In contrast, we use as the supervisory signal the strong causal ordering constraints that relate states of objects and the state-modifying actions.

Action recognition and localization. The problem of detecting, classifying and localizing human actions has been extensively addressed by methods exploring motion and temporal evolution of appearance in a video. Models for action recognition typically operate on short video clips trimmed to encompass a single action. Such models employ a mix of 2D and 3D convolutions [14, 25, 26, 61] or transformers and temporal attention [9, 23, 68]. Action localization methods often generate action proposals in the

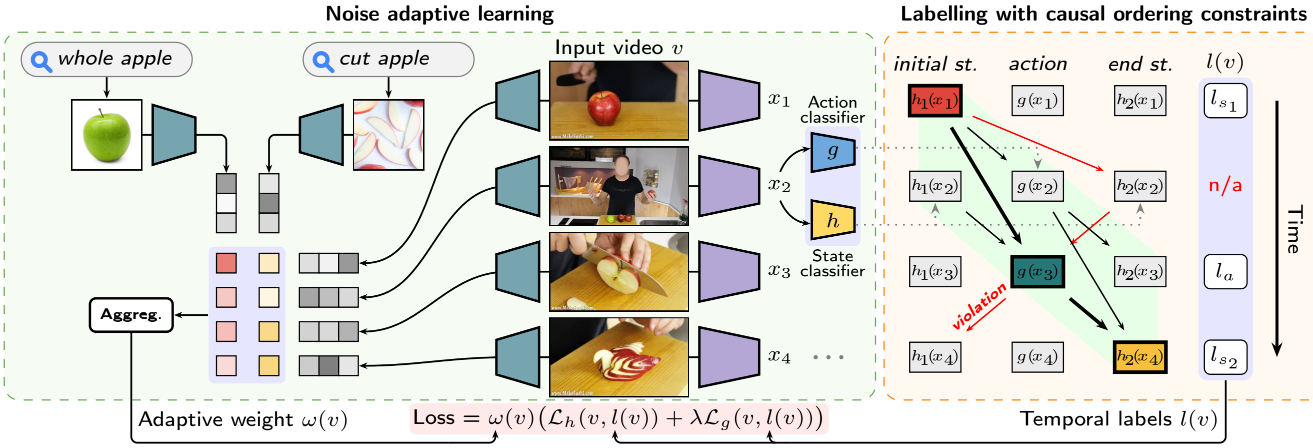


Figure 2. **Model overview.** Given a set of input *noisy untrimmed videos* from the web depicting a state-changing action (here *cutting apple*) our approach learns action classifier g and object state classifier h that output temporal labels l of the input videos with temporal locations of *initial object state* $\rightarrow$ *manipulating action* $\rightarrow$ *end object state* that satisfy the *causal ordering constraint*. This is achieved by minimizing a new noise adaptive learning objective that downweights irrelevant videos with adaptive weight ω measuring similarity to a small number of exemplar images. The learning proceeds by iteratively **(i)** learning action and state classifiers, g and h, given the current labels l of the input videos and **(ii)** finding the labels l of the videos that respects the causal ordering constraints.

temporal domain using special modules such as graph neural networks [6, 13, 36, 49, 66]. However, such methods typically require video annotations in terms of temporal action boundaries for training. Our proposed methods does not require temporal supervision. It uses changes in object states as a guidance for action localization.

Object states and action video datasets. Most existing video action recognition datasets primarily contain state-preserving actions such as *dancing* or *playing a flute* [14, 34, 57]. EPIC-KITCHENS [17], Breakfast [33], CrossTask [73] or COIN [59] datasets provide action sequence and object annotations for each video, but do not provide annotations related to changes of object states. HowTo100M [45], YouCook2 [71] and RareAct [42] datasets contain videos with state-changing actions; however, they also do not provide clearly defined object-state annotations. Closely related to us, Alayrac *et al*. [3] introduced an annotated video dataset of state-changing actions. However, this dataset was carefully curated to ensure that each video contains the action and object state change of interest. Consequently, their dataset is small scale and contains only seven object-action classes with only tens of videos for each class. The Task-Fluent dataset [38] features several state-changing actions but is restricted to only 809 ego-centric videos. In contrast, our dataset is 54× and 42× larger than the datasets of [3] and [38], respectively, and contains untrimmed videos of a large variety of 44 different object-action classes. Concurrent to our work, a recently collected EGO4D [30] dataset contains 3,025 hours of ego-centric videos and also provides state change and action annotations. Our dataset is of comparable size but has a different focus on untrimmed videos from the web.

3. Learning of actions and object states from untrimmed web videos

We are given a set of web videos $v \in \mathcal{V}$ of arbitrarily length likely to depict a common state-modifying action applied on an object. For example $\mathcal{V}$ can be a collection of birthday celebration videos, which are all likely to contain people *blowing out candles* (*i.e*. action) and changing the state of the candles from *lighted* (*i.e*. initial state) to *extinguished* (*i.e*. final state).

Given this, our goal is twofold: **(i)** learning an action classifier g that can recognize the action of interest and **(ii)** learning a state classifier h categorizing the modified object into an *initial state* and an *end state*. We seek to do so without access to ground truth labels for the actions nor the object states. Instead, we design an approach that relies on the supervision provided by the causality of time: the action should appear between the two object states. In addition, we show that a handful of additional labelled exemplar images depicting the two object states help to make our approach significantly more robust to the noise in the training data via our new noise adaptive learning objective.

In detail, the proposed learning procedure, illustrated in Figure 2, optimizes the following objective:

$$\min_{g,h} \sum_{v \in \mathcal{V}} \mathcal{L}_{(g,h)}(v, l(v)) \quad \text{(Sec. 3.1)}, \tag{1}$$

where h and g are the learnt state and action classifiers, respectively, $\mathcal{L}$ is a loss function adapted to the noisy nature of web videos and $l(v)$ are labels for both the action and state temporal positions within the video v. Since these labels are not given in advance, we instead estimate it on the fly during the optimization procedure via the following maximization:

$$l(v) = \underset{l \in \mathcal{D}_v}{\arg\max}\, S_{(g,h)}(v, l) \quad \text{(Sec. 3.2)}, \tag{2}$$

where S is a scoring function that depends on the current action and state classifiers g and h. $\mathcal{D}_v$ is the set of labels that respect our temporal causality constraints. The learning proceeds by iteratively **(i)** learning action and state classifiers, g and h, given the current labels l of the input videos (Eq. (1) and Sec. 3.1) and **(ii)** finding the labels l of the videos that respects the causal ordering constraints given the current action and state classifiers, g and h (Eq. (2) and Sec. 3.2). Details about these two steps are provided next.

3.1. Noise adaptive learning objective

In this section, we describe the loss function $\mathcal{L}$ from Equation (1). Each video v is represented by a sequence of T_v d-dimensional visual features: $v = \{x_t\}_{t=1}^{T_v}$. Each $x_t \in \mathbb{R}^d$ describes a temporal segment one second long of the original video. Here, we assume that the labels $l(v)$ are known for all videos, *i.e.* the temporal position of the action $l_a(v) \in [\![1, T_v]\!]$ as well as the temporal positions of the initial state $l_{s_1}(v) \in [\![1, T_v]\!]$ and the end state $l_{s_2}(v) \in [\![1, T_v]\!]$ (see Section 3.2 for details about how $l(v)$ is obtained).

Action and state classifiers. The goal here is to learn the action and state classifiers, g and h, given the labels l. The action classifier g takes as input a visual feature $x \in \mathbb{R}^d$ and outputs a confidence score $g(x) \in [0, 1]$ that the feature depicts the action of interest. Similarly, the state classifier h takes as input the visual feature x and outputs two scores $h_1(x),\ h_2(x) \in [0, 1]$ giving an estimate of probability that the feature corresponds to the initial and the end state.

Loss definition. Formally, the loss function $\mathcal{L}_{(g,h)}$ of a video v and its associated labels $l(v)$ is a weighted sum of losses $\mathcal{L}_g$ for the action and $\mathcal{L}_h$ for the states:

$$\mathcal{L}_{(g,h)}(v, l(v)) = \omega(v)\big(\mathcal{L}_h(v, l(v)) + \lambda \mathcal{L}_g(v, l(v))\big) \tag{3}$$

where λ controls the relative importance of the two partial losses, and g and h are the action and state classifiers, respectively, that are being learnt. Given the noisy nature of the dataset of untrimmed videos obtained from the web, we weight each video's contribution to the overall loss function by a scalar weight $\omega(v)$, which is deduced from comparing the video frames to a small set of exemplar images (Figure 2, bottom left) and described later.

The action and state losses in Eq. (3) are cross-entropy losses applied on the output of the classifiers as:

$$\mathcal{L}_g(v, l(v)) = -\mu \sum_{t \in \mathcal{A}_v^P} \log g(x_t) - \sum_{t \in \mathcal{A}_v^N} \log\big(1 - g(x_t)\big)$$
$$\mathcal{L}_h(v, l(v)) = -\sum_{t \in \mathcal{S}_v^1} \log h_1(x_t) - \sum_{t \in \mathcal{S}_v^2} \log h_2(v_t) \tag{4}$$

where $\mathcal{S}_v^1$, $\mathcal{S}_v^2$, $\mathcal{A}_v^P$ are sets of positive examples deduced from $l(v)$ where the model is expected to predict the initial state, the end state and the action, respectively. The set $\mathcal{A}_v^N$ contains negative examples where the model is expected to produce the *no-action* label. We describe how these sets are deduced from the current labels $l(v)$ of the video below. The parameter μ is the relative weight between the action/no-action class.

Sampling of positive examples. All the positive sets $\mathcal{S}_v^1$, $\mathcal{S}_v^2$, $\mathcal{A}_v^P$ are sampled in the same way and are directly obtained from labels $l(v)$. They all contain feature indices within a temporal window centered on the currently estimated locations of the initial state $l_{s_1}(v)$, end state $l_{s_2}(v)$ and the action $l_a(v)$. Formally, the set of positive examples for the initial state $t \in \mathcal{S}_v^1$ is defined as:

$$\mathcal{S}_v^1 = \big\{t : |t - l_{s_1}(v)| \le \delta,\ 1 \le t \le T_v\big\} \tag{5}$$

where $l_{s_1}(v)$ is the currently estimated position of the initial state in video v, T_v is the length of the video, and δ is a hyper-parameter defining the number of the neighbouring features considered as positive. The intuition is that we wish to consider as positives several temporally nearby examples (within the temporal window defined by δ) as they are likely to also contain the initial object state. The sets of positive examples for the end state and action, $\mathcal{S}_v^2$ and $\mathcal{A}_v^P$, are defined analogously.

Sampling of *no-action* examples. There are various ways of sampling the set $\mathcal{A}_v^N$ of *no-action* examples. Considering all negatives in the video, $\mathcal{A}_v^N = \{t : t \notin \mathcal{A}_v^P\}$, is impractical due to class imbalance which is a) directly dependent on the length of the video and b) extremely large with ratios that can exceed 1 to 100 in long videos. Instead, we opt for defining $\mathcal{A}_v^N$ as a set of video feature indices at a given distance κ from the location $t' \in \mathcal{A}_v^P$ of the positive example:

$$\mathcal{A}_v^N = \big\{t : t' \in \mathcal{A}_v^P,\ |t - t'| = \kappa,\ 1 \le t \le T_v\big\}. \tag{6}$$

The intuition is that, for appropriate κ, set $\mathcal{A}_v^N$ will contain *hard* negatives, that are visually related to the correct action but yet negative. If κ is too small, $\mathcal{A}_v^N$ will contain positive examples, which will harm training the action classifier. On the other hand, if κ is too large, $\mathcal{A}_v^N$ can contain unrelated (easy to discriminate) actions from the rest of the video. In Section 5, we ablate the choice of κ and show that this definition of negatives for the action classifier is crucial for obtaining good performance compared to randomly sampling the positions of the negatives. Lastly, we note that there can be positions in a video that are not in any of the four $\mathcal{S}_v^1$, $\mathcal{S}_v^2$, $\mathcal{A}_v^P$, $\mathcal{A}_v^N$ sets. Actually, in the case of longer videos, most of the segments are without any label and thus do not contribute to the loss.

Noise adaptive weighting from a few exemplar images. As our training videos are obtained automatically from the web without any manual curation, a large proportion of videos may contain unrelated content and thus harm the performance of the model. To address this issue, we download a small number of images (up to five) via Google Image search containing the object of interest in both initial and end states. Then we use a pre-trained visual model applied in a zero-shot manner together with the causal ordering constraint to compute video relevance score r_v as follows:

$$r_v = \max_{t<t'} \sum_{e_1 \in \mathcal{E}_1} \mathrm{sim}\left(e_1, v_t\right) \sum_{e_2 \in \mathcal{E}_2} \mathrm{sim}\left(e_2, v_{t'}\right) \quad (7)$$

where $\mathcal{E}_1$, $\mathcal{E}_2$ are sets of exemplar images representing the initial and the end state, respectively, and $\mathrm{sim}\left(e, v_t\right)$ is the similarity between the exemplar image e and the video feature at the t-th temporal location of video v computed as cosine similarity of features extracted by a pre-trained visual model. We use this relevance score to weigh the contribution of each video in the loss function using the following weight:

$$\omega(v) = \sigma\left(\frac{r_v - \theta}{\tau}\right) \quad (8)$$

where σ is sigmoid function, τ is a temperature hyper-parameter and θ is a centering hyper-parameter. The relevance weight $\omega(v)$ varies between 0 and 1. The weight is close to 0 for videos that do not have any frames similar to the object state exemplar images e satisfying the causal ordering constraint. On the other hand, the weight is close to 1 for videos, which have frames with high similarity to exemplar images e and which satisfy the causal ordering constraint. As a result, this weight is effective in suppressing irrelevant videos during the learning process. Note that we do not use exemplar action images as we found their collection at scale to be problematic. We select θ for each object-action category independently because the relevance scores r_v vary greatly depending on the exemplar images or the video content. We use θ that minimizes intra-class variance of the relevance scores. More details are given in the appendix [1].

3.2. Labelling with causal ordering constraints

In this section, we explain how labels $l(v)$ identifying the best action and object state locations in video v are automatically obtained given the current, possibly sub-optimal, action and state classifiers, g and h. More formally, we assume we are given fixed classifiers g for an appearance-changing action and h for the manipulated object's states. We are also given a video v containing the action exerted upon the object with high probability. Then to compute the most likely location of the action $l_a(v)$, the initial state $l_{s_1}(v)$ and the end state $l_{s_2}(v)$, we employ predictions of the current action g and object state classifiers h as follows:

$$l(v) = \underset{l \in \mathcal{D}_v}{\arg\max}\, h_1(x_{l_{s_1}}) \cdot g(x_{l_a}) \cdot h_2(x_{l_{s_2}}) \quad (9)$$

where $\mathcal{D}_v$ is a set of all possible locations of the action and the object states satisfying the *causal ordering constraint*, $h_1(x_{l_{s_1}})$ is the output of the initial state classifier h_1 at temporal location l_{s_1} in video v, $h_2(x_{l_{s_2}})$ is the output of the end state classifier h_2 at temporal location l_{s_2}, and $g(x_{l_a})$ is the output of action classifier g at temporal location l_a. In other words, the goal is to identify object state and action locations in the video that satisfy the causal ordering constraint and maximize the product of output scores of the state and action classifiers as given in Eq. (9).

Causal ordering constraint. The causal ordering constraint employed in Eq. (9) is motivated by the fact that many object-modifying actions cannot be physically reversed, *e.g. cut apple*. Also, many object-modifying actions are commonly performed only one way, even if the other direction is physically possible, *e.g. clean shoes*. Thus we restrict the set of permissible locations of actions and states $\mathcal{D}_v$ to follow the order of initial object state $\rightarrow$ manipulating action $\rightarrow$ end object state. Formally, we define the set $\mathcal{D}_v$ of labels satisfying this constraint as

$$\mathcal{D}_v = \left\{ l : 1 \leq l_{s_1} < l_a < l_{s_2} \leq T_v \right\} \quad (10)$$

where T_v is the length of the video v and l_{s_1}, l_a, l_{s_2} are the temporal positions of the initial state, action and the end state, respectively. In untrimmed videos from the web, it is common to have multiple instances of objects of interest or distracting objects in the same video. The same is true for actions. The ordering constraint pinpoints the most prominent instance of the object and action in each video. Other instances are ignored and treated as background.

4. ChangeIt: a state-changing actions dataset

Our goal is to automatically learn the different states of objects together with the state-modifying actions without the need for manual curation of videos. To this end, we collect a new large-scale dataset of more than 34,000 in-the-wild untrimmed videos (more than 2600 hours of video) covering 44 various state-changing actions. Our state-changing actions depict a wide range of human activities such as: *cleaning shoes* (initial state: dirty shoes, action: cleaning, final state: clean shoes), *cut avocado* (initial state: entire avocado, action: cutting, final state: cut avocado) or *gift wrapping* (initial state: unwrapped gift, action: wrapping, final state: wrapped gift). We emphasize the training split of our **ChangeIt** dataset is collected in such a way that only a name of an action is needed to be specified. No other manual selection, acquisition or annotation is required. Next we describe our dataset collection process and provide detailed statistics of **ChangeIt**.

Dataset collection. First, we select a set of state-changing actions. Such actions imply a modification of the appearance of an object through manipulation. We restrict ourselves mostly to irreversible actions in order to eliminate scenarios where two actions manipulate an object from an initial state back to the same initial state via an end state, such as *open* and *close a door*. However, we allow actions such as *clean shoes* as these are not considered immediately reversible because we do not expect the shoes to be immediately dirty after being cleaned. We also require the actions to be sufficiently represented on YouTube.

Given these conditions, we devise a set of 44 state-changing actions. Note that some of the classes contain similar or complementary actions such as *cut* and *peel an avocado*. For each action, we query YouTube with queries such as *"How to clean shoes?"* and download up to two thousand retrieved videos. We exclude videos longer than 15 minutes and obtain 34,428 videos with a total duration of 2,642 hours and an average video duration of 4.6 minutes. We have manually annotated a small fraction of videos for evaluation with the following labels: *background*, *initial state*, *action*, *end state*. In total we have annotated 667 videos summing up to 48 video hours and yielding 15 video samples per state-changing action on average. Please see the appendix [1] for more details on the action selection, annotation, and additional dataset statistics.

5. Experiments

In this section, we first describe the model architecture and training regime (Section 5.1). Then we describe the datasets we test on, the evaluation process, and the metrics we report (Section 5.2). We ablate our most important design decisions and show their benefit in Section 5.3. Finally, we compare our method with related work and various baselines and show qualitative results in Section 5.4.

5.1. Implementation details

Architecture. Our definition of the model learning procedure (Section 3) allows us to use any differentiable temporal video classifier for g and h. We choose both classifiers to be applied on the same visual features and share the same architecture: a two-layer MLP with hidden dimension of 512 and ReLU activation function. We train separate classifiers for every dataset category, thus the action classifier outputs a single scalar followed by the sigmoid activation function, the state classifier outputs two scalar followed by the softmax activation function. The feature extractors operate on original videos and downsample the temporal resolution of the video features x_t into one frame per second. For the feature extractors, we use 2D ResNeXT pre-trained on ImageNet-21K [41] and 3D TSM ResNet50 pre-trained on HowTo100M and AudioSet [4]. The 2D and 3D features are concatenated prior to be fed to the classifiers. We initialize the MLPs randomly [31]. We do not back-propagate gradients into the feature extractors.

Training regime. During training, we sample a batch of videos and compute action and states locations for each video in the batch (Eq. (9)). Then we compute gradients of the loss function $\mathcal{L}_{(g,h)}(v)$ with respect to the model parameters and perform one step of gradient descent with a momentum. We alternate these steps for 100 epochs. Additional details with all hyper-parameter values and data preprocessing steps are in the appendix [1].

5.2. Evaluation protocol and metrics

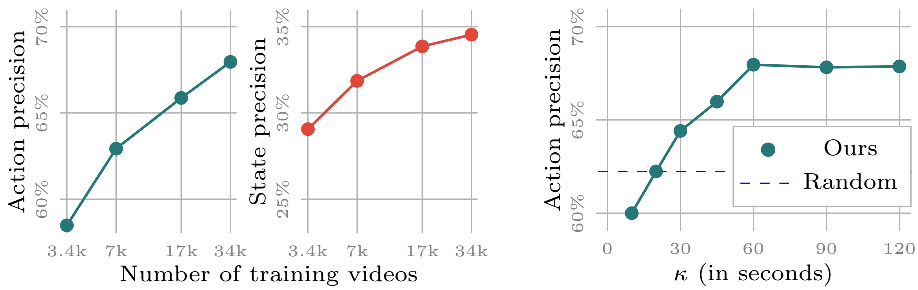
We evaluate on both our new **ChangeIt** dataset and on the dataset of Alayrac *et al*. [3]. The first one depicting noisy untrimmed videos and the latter one depicting short curated trimmed videos. For each video we predict the location of the action and the initial and end states by Equation (9) using predictions of the trained classifiers. For Alayrac *et al*. dataset, we average predictions for frames corresponding to a single so-called *tracklet* and apply Equation (9) on the averaged predictions only. In all experiments including ablations, we report performance of the best performing model weights from training averaged over three runs.

We follow the related work [3] with metrics and report precision for both the action and the states. For a given video, action precision is either one or zero depending on whether the predicted action location is within the ground truth interval. For state precision, a value of 0.5 is also possible if only one of the two state locations matches the ground truth. The video-level metrics are then averaged over all videos in a category, and finally averaged over all categories to suppress effects of differences in distributions of videos throughout the categories.

5.3. Ablations

In this section, we ablate the key components of our model and show their benefit. We investigate the effect of dataset size, model depth and input data augmentation on performance. We also show the importance of sampling action negatives and noise adaptive video weighting during training. All experiments are performed on the full **ChangeIt** dataset except for the dataset size ablation.

Dataset size matters. We train our model on various fractions of the dataset to investigate the effect of the dataset size on performance. In each category, we sort videos according to their relevance score r_v (Eq. (7)) and train our model with the top scoring 5%, 10%, 20%, 50% and 100% of videos. Figure 3a shows the effect of varying the dataset size on action and state precision. There is a clear improvement with increasing dataset size. Also, we do not observe performance saturation indicating further improvements could be achieved with even larger datasets. This observation is even more interesting when we consider that many low scoring videos may not contain the action or even the object of interest, yet the model improves even in this low signal-to-noise ratio setting.



(a) Dataset size **(b) Negative sampling**

Method	St prec.	Ac prec.
a. Ours	**0.35**	**0.68**
b. w/o noise adapt.	0.34	0.64
c. w/o multi-layer	0.34	0.61
d. w/o data augment.	0.33	0.66

(c) Selected variants of our model

Figure 3. **Ablations on the ChangeIt dataset.** (a) State and action precision with the increasing size of the training dataset. (b) Action precision as a function of the distance κ of sampled negatives from the positives compared with random negative sampling (Random). (c) Ablations of different components of our model.

Sampling action negatives. In our experiments, we show that the strategy for sampling action negatives $\mathcal{A}_N$ is an important design choice significantly affecting the model's action localization performance. When the negatives are sampled uniformly at random, as shown in Figure 3b in blue, we see a drop from 68% to 62% in action precision compared to our method of using negatives that are at a fixed distance κ from the positives. We also test different values of κ (Figure 3b). We can see the best performance is achieved using $\kappa \geq 60$. We hypothesize that by using smaller values of κ the action negatives can still contain the action which hinders model performance. Nonetheless, most κ values still outperform the random sampling. Increasing κ further yields only a minor change in the precision as most of the videos are 1-5 minutes long and thus the negatives often lay outside of the video. In such cases, the first or the last frames of the video are taken as negatives.

Noise adaptive weighting. We test the added benefit of using video relevance scores to weigh individual videos in the batch. These relevance scores incorporate the only truly manual data collection in the whole method, even if very minor as only a handful of images are required. Results without the noise adaptive video weighting are shown in Figure 3c (row **b. w/o noise adapt**) and show a clear drop in performance, especially for actions compared to the full method (row **a. Ours**).

Other ablations. We test the benefits of our two-layer MLP on top of a feature extractor compared to a single linear layer used in related work [3]. In Figure 3c, we see a clear benefit of using the two-layer MLP **a. Ours** compared to a linear classifier **c. w/o multi-layer** in the same set-up suggesting that a linear classifier used by Alayrac *et al.* [3] for it's closed-form solution is insufficient to discriminate between states and actions. We also add augmentation for the input videos and show the benefit of this standard practice in neural network training in our setup (Figure 3c, row **d. w/o data augment.**). Additionally, in the appendix [1], we show the effect of different feature extractor backbones on the final performance.

5.4. Comparison with the state-of-the-art

Compared methods. We compare our method to several strong baselines described next. **(a) Alayrac *et al.* [3].** We compare results with the state-of-the-art approach for learning object states and actions from video by Alayrac *et al.* [3], which learns a linear classifier on fixed video features using discriminative clustering [5]. As we notice this method could be unstable, we report the best numbers reached during the course of optimization. **(b) CLIP [53].** We compare to the zero-shot CLIP approach [53] that has demonstrated strong results on a large variety of recognition tasks and thus presents a strong baseline. We obtain the state and action classifiers by projecting the textual descriptions of our action and object state classes into the joint text-image space. We employ prompt engineering by producing multiple text descriptions for each state and action and report the performance of the best one. To make the comparison as fair as possible, we employ the causal ordering constraint for computing the state and action precision as in Equation (9). **(c) MIL-NCE S3D [43]** Analogously to CLIP, we also compare to the zero-shot video-based S3D model trained on the HowTo100M dataset [45] using MIL-NCE loss [43]. We use the same evaluation procedure as for CLIP with the same text descriptions and the causal ordering constraint employed. **(d) Image examples.** We use the images gathered for the noise adaptive weighting and measure their similarity to individual video frames in a feature space. We use the causal ordering constraint for the computation of the state precision metric. **(e) Random.** We also report chance performance with the state ordering constraint employed.

Comparison on the dataset of Alayrac *et al.* In Table 1 we report performance on the dataset from Alayrac *et al.* [3] containing curated trimmed videos of seven action-object classes. Note that some dataset classes contain only tens of videos thus we use additional temporal attention using $sim(\mathcal{E}_*, v_t) \cdot h_*(x_t)$ instead of $h_*(x_t)$ in Equation 9. Without it, our method can focus on different temporally consistent actions that are also present in the same video, *e.g. jack*

Method	St prec.	Ac prec.
Random	0.09	0.22
CLIP ViT-L/14 [53]	0.42	0.42
Alayrac *et al.* [3]†	0.48	0.55
Ours	**0.49**	**0.58**

† Trained with known object bounding boxes.

Table 1. Comparison to the state-of-the-art approach [3] on their own dataset.

Method	St prec.	Ac prec.
(e) Random w/ constraint	0.15	0.41
(d) Image examples	0.29	-
(c) MIL-NCE S3D [43]	0.27	0.50
(b) CLIP ViT-L/14 [53]	0.30	0.63
(a) Alayrac *et al.* [3]	0.30	0.59
Ours	**0.35**	**0.68**

Table 2. Comparison to state-of-the-art on our **ChangeIt** dataset.

up a car instead of *remove a wheel*. The results in Table 1 demonstrate the benefits of our approach over [3], despite fact that [3] uses much stronger supervision in the form of a pre-trained object detector for each objects.

Results on our new ChangeIt dataset. In Table 2 we report quantitative results on our much larger **ChangeIt** dataset that contains long untrimmed uncurated videos. We observe that the zero-shot CLIP method **(b)** and the image-based approach **(c)** match the state-of-the-art approach of **(a)** Alayrac *et al.* [3]. Our method produces significantly better results and outperforms the state-of-the-art approach of Alayrac *et al.* [3] as well as the other baselines, demonstrating the benefits of our approach on noisy untrimmed videos. The full set of per-class results on the entire set of 44 classes is in the appendix [1].

Qualitative results. Qualitative results are shown in Figures 1 and 4. They demonstrate the ability of our approach to learn object state and action classifiers from long uncurated videos from the web. Additional qualitative results for a range of object-action classes are in the appendix [1].

Limitations and societal impact. Our approach has currently three main limitations. First, it relies on training videos being available on the web. This may not be the case for all state-modifying actions. For example, some common or uninteresting daily actions (such as *open a fridge*) may not be frequently captured and uploaded to Youtube. Second, in some cases the appearance variation of the object states is too large, *e.g. remove weed*, to be learnt in a fully unsupervised manner. Finally, currently we learn a separate model for each class from a set of videos for that class. Enabling sharing information during learning among models of related classes, *e.g. cut apple* and *cut avocado*, remains an interesting open problem. Further discussion about potential negative societal impact of our work is provided in the appendix [1].

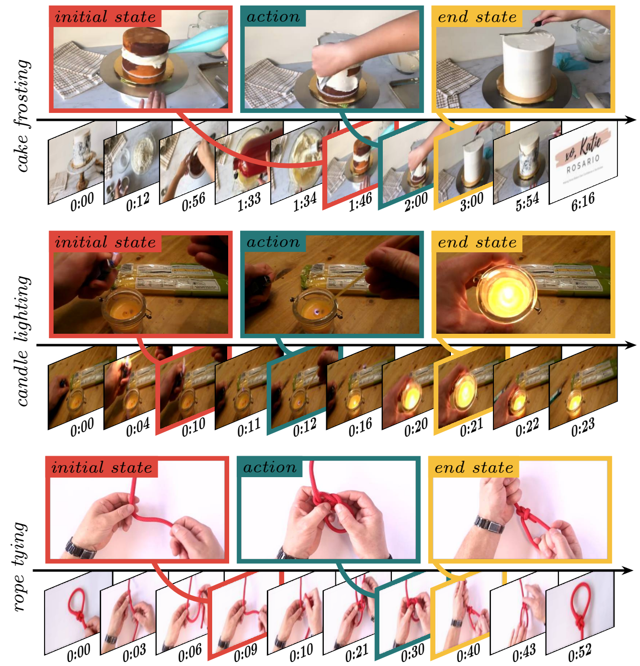


Figure 4. **Predicted object-state and action frames** (top 3 frames in each example) and their temporal localization on the video timeline (bottom). Note how the object states and the state-modifying actions are temporally localized in long uncurated videos from the web. See additional examples in the appendix [1].

6. Conclusion

We have developed a new approach for learning object-states and state-changing actions from noisy untrimmed videos from the web. Our approach relies on a novel noise-adaptive learning objective supervised by exemplar images together with causal ordering constraints temporally relating object changes and actions in videos. We have validated our approach on an existing dataset of state-modifying actions as well as our newly collected dataset of more than 2600 hours of videos from the web, demonstrating significant improvements compared to the state-of-the-art. This work opens-up the possibility of large-scale automatic learning of causal object-action relations for embodied video understanding and robotics.

Acknowledgements. This work was partly supported by the European Regional Development Fund under the project IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15_003/0000468), the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90140), the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR19-P3IA-0001 (PRAIRIE 3IA Institute), and Louis Vuitton ENS Chair on Artificial Intelligence. We would like to also thank Kateřina Součková and Lukáš Kořínek for their help with the dataset.

References

[1] Supplementary material (appendix) for the paper. http://arxiv.org/abs/2203.11637. 5, 6, 7, 8

[2] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Ivan Laptev, Josef Sivic, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *CVPR*, 2016. 2

[3] Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Simon Lacoste-Julien. Joint discovery of object states and manipulation actions. In *ICCV*, 2017. 1, 2, 3, 6, 7, 8

[4] Jean-Baptiste Alayrac, Adria Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. In *NeurIPS*, 2020. 2, 6

[5] Francis Bach and Zaïd Harchaoui. DIFFRAC: a discriminative and flexible framework for clustering. In *NIPS*, 2007. 7

[6] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. Boundary content graph neural network for temporal action proposal generation. In *ECCV*, 2020. 3

[7] Fernando Basura, Efstratios Gavves, José M. Oramas, Amir Ghodrati, and Tinne Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, 2015. 2

[8] Gedas Bertasius and Lorenzo Torresani. Cobe: Contextualized object embeddings from narrated instructional video. In *NeurIPS*, 2020. 2

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

[10] Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Weakly supervised action labeling in videos under ordering constraints. In *ECCV*, 2014. 2

[11] Piotr Bojanowski, Rémi Lajugie, Edouard Grave, Francis Bach, Ivan Laptev, Jean Ponce, and Cordelia Schmid. Weakly-supervised alignment of video with text. In *ICCV*, 2015. 2

[12] T. F. Brady, T. Konkle, A. Oliva, and G. A. Alvarez. Detecting changes in real-world objects: The relationship between visual long-term memory and change blindness. *Communicative and Integrative Biology*, 2006. 1

[13] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. Sst: Single-stream temporal action proposals. In *CVPR*, 2017. 3

[14] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 2, 3

[15] Chien-Yi Chang, De-An Huang, Danfei Xu, Ehsan Adeli, Li Fei-Fei, and Juan Carlos Niebles. Procedure planning in instructional videos. In *ECCV*, 2020. 2

[16] Mithun Chowdhury, Panda Rameswar, Evangelos Papalexakis, and Amit Roy-Chowdhury. Webly supervised joint embedding for cross-modal image-text retrieval. In *ACM International Conference on Multimedia*, 2018. 2

[17] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018. 3

[18] Dima Damen, Teesid Leelasawassuk, Osian Haines, Andrew Calway, and Walterio Mayol-Cuevas. You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video. In *BMVA*, 2014. 1, 2

[19] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *CVPR*, 2019. 2

[20] Hazel Doughty, Dima Damen, and Walterio Mayol-Cuevas. Who's better? who's best? pairwise deep ranking for skill determination. In *CVPR*, 2018. 2

[21] Hazel Doughty, Ivan Laptev, Walterio Mayol-Cuevas, and Dima Damen. Action modifiers: Learning from adverbs in instructional videos. In *CVPR*, 2020. 2

[22] Dave Epstein, Jiajun Wu, Cordelia Schmid, and Chen Sun. Learning temporal dynamics from cycles in narrated video. In *ICCV*, 2021. 2

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

[24] Alireza Fathi and James M. Rehg. Modeling actions through state changes. In *CVPR*, 2013. 2

[25] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, 2020. 2

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

[27] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. 2021. 2

[28] Yunchao Gong, Qifa Ke, Michael Isard, and Svetlana Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *IJCV*, 2014. 2

[29] Yunchao Gong, Liwei Wang, Micah Hodosh, Julia Hockenmaier, and Svetlana Lazebnik. Improving image-sentence embeddings using large weakly annotated photo collections. In *ECCV*, 2014. 2

[30] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo,

Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4D: Around the world in 3,000 hours of egocentric video. 2021. 3

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 6

[32] Phillip Isola, Joseph J. Lim, and Edward H. Adelson. Discovering states and transformations in image collections. In *CVPR*, 2015. 2

[33] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *CVPR*, 2014. 3

[34] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011. 3

[35] Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. Unsupervised learning of action classes with continuous temporal embedding. In *CVPR*, 2019. 2

[36] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *ICCV*, 2019. 3

[37] 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. 2

[38] Yang Liu, Ping Wei, and Song-Chun Zhu. Jointly recognizing object fluents and tasks in egocentric videos. In *ICCV*, 2017. 2, 3

[39] Pauline Luc, Aidan Clark, Sander Dieleman, Diego de Las Casas, Yotam Doron, Albin Cassirer, and Karen Simonyan. Transformation-based adversarial video prediction on large-scale data. *arXiv:2003.04035*, 2020. 2

[40] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016. 2

[41] Pascal Mettes, Dennis C. Koelma, and Cees G. M. Snoek. Shuffled imagenet banks for video event detection and search. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2020. 6

[42] Antoine Miech, Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Andrew Zisserman. Rareact: A video dataset of unusual interactions. *arxiv:2008.01018*, 2020. 3

[43] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-End Learning of Visual Representations from Uncurated Instructional Videos. In *CVPR*, 2019. 2, 7, 8

[44] Antoine Miech, Ivan Laptev, Josef Sivic, Heng Wang, Lorenzo Torresani, and Du Tran. Leveraging the present to anticipate the future in videos. In *CVPR Workshops*, 2019. 2

[45] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100M: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019. 2, 3, 7

[46] Ishan Misra, Abhinav Gupta, and Martial Hebert. From red wine to red tomato: Composition with context. In *CVPR*, 2017. 2

[47] Muhammad Ferjad Naeem, Yongqin Xian, Federico Tombari, and Zeynep Akata. Learning graph embeddings for compositional zero-shot learning. In *CVPR*, 2021. 2

[48] Tushar Nagarajan and Kristen Grauman. Attributes as operators: Factorizing unseen attribute-object compositions. In *ECCV*, 2018. 2

[49] Megha Nawhal and Greg Mori. Activity graph transformer for temporal action localization. *arXiv:2101.08540*, 2021. 3

[50] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, 2016. 2

[51] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015. 2

[52] Senthil Purushwalkam, Maximilian Nickel, Abhinav Gupta, and Marc'Aurelio Ranzato. Task-driven modular networks for zero-shot compositional learning. In *ICCV*, 2019. 2

[53] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 2, 7, 8

[54] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv:1412.6604*, 2014. 2

[55] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *IJCV*, 2017. 2

[56] Fadime Sener and Angela Yao. Unsupervised learning and segmentation of complex activities from video. In *CVPR*, 2018. 2

[57] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv:1212.0402*, 2012. 3

[58] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015. 2

[59] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *CVPR*, 2019. 3

[60] Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *PAMI*, 2018. 2

[61] Limin Wang, Yuanjun Xiong, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 2

[62] Xiaolong Wang, Ali Farhadi, and Abhinav Gupta. Actions ~ transformations. In *CVPR*, 2016. 2

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

[64] Michael Wray, Diane Larlus, Gabriela Csurka, and Dima Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *ICCV*, 2019. 2

[65] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016. 2

[66] Mengmeng Xu, Chen Zhao, David S. Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *CVPR*, 2020. 3

[67] Ran Xu, Caiming Xiong, Wei Chen, and Jason J Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI*, 2015. 2

[68] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. *arXiv:2201.04288*, 2022. 2

[69] Muli Yang, Cheng Deng, Junchi Yan, Xianglong Liu, and Dacheng Tao. Learning unseen concepts via hierarchical decomposition and composition. In *CVPR*, 2020. 2

[70] Da Zhang, Xiyang Dai, and Yuan-Fang Wang. Metal: Minimum effort temporal activity localization in untrimmed videos. In *CVPR*, 2020. 2

[71] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 2, 3

[72] Yipin Zhou and Tamara L Berg. Learning temporal transformations from time-lapse videos. In *ECCV*, 2016. 2

[73] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *CVPR*, 2019. 3

[74] Dimitri Zhukov, Jean-Baptiste Alayrac, Ivan Laptev, and Josef Sivic. Learning actionness via long-range temporal order verification. In *ECCV*, 2020. 2