

Joint Discovery of Object States and Manipulation Actions

Jean-Baptiste Alayrac^{*†}Josef Sivic^{*†‡}Ivan Laptev^{*†}Simon Lacoste-Julien[§]

Abstract

Many human activities involve object manipulations aiming to modify the object state. Examples of common state changes include full/empty bottle, open/closed door, and attached/detached car wheel. In this work, we seek to automatically discover the states of objects and the associated manipulation actions. Given a set of videos for a particular task, we propose a joint model that learns to identify object states and to localize state-modifying actions. Our model is formulated as a discriminative clustering cost with constraints. We assume a consistent temporal order for the changes in object states and manipulation actions, and introduce new optimization techniques to learn model parameters without additional supervision. We demonstrate successful discovery of seven manipulation actions and corresponding object states on a new dataset of videos depicting real-life object manipulations. We show that our joint formulation results in an improvement of object state discovery by action recognition and vice versa.

1. Introduction

Many of our activities involve changes in object states. We need to open a book to read it, to cut bread before eating it and to lighten candles before taking out a birthday cake. Transitions of object states are often coupled with particular manipulation actions (open, cut, lighten). Moreover, the success of an action is often signified by reaching the desired state of an object (whipped cream, ironed shirt) and avoiding other states (burned shirt). Recognizing object states and manipulation actions is, hence, expected to become a key component of future systems such as wearable automatic assistants or home robots helping people in their daily tasks.

Human visual system can easily distinguish different states of objects, such as open/closed bottle or full/empty coffee cup [8]. Automatic recognition of object states and

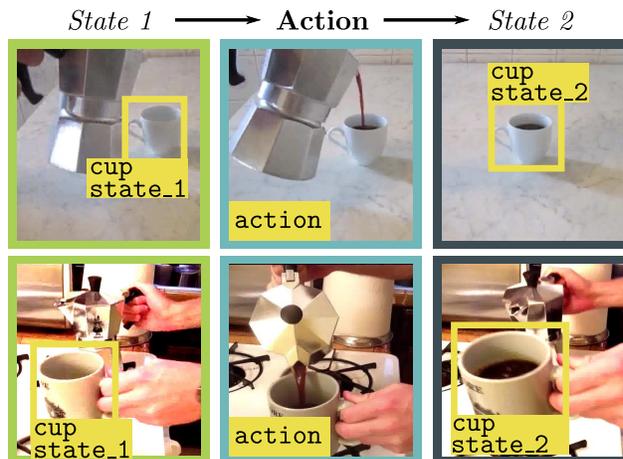


Figure 1: We automatically discover object states such as empty/full coffee cup along with their corresponding manipulation actions by observing people interacting with the objects.

state changes, however, presents challenges as it requires distinguishing subtle changes in object appearance such as the presence of a cap on the bottle or screws on the car tire. Despite much work on object recognition and localization, recognition of object states has received only limited attention in computer vision [22].

One solution to recognizing object states would be to manually annotate states for different objects, and treat the problem as a supervised fine-grained object classification task [14, 15]. This approach, however, presents two problems. First, we would have to decide *a priori* on the set of state labels for each object, which can be ambiguous and not suitable for future tasks. Second, for each label we would need to collect a large number of examples, which can be very costly.

In this paper we propose to *discover* object states directly from videos with object manipulations. As state changes are often caused by specific actions, we attempt to jointly discover object states and corresponding manipulations. In our setup we assume that two distinct object states are temporally separated by a manipulation action. For example, the empty and full states of a coffee cup are separated by the “pouring coffee” action, as shown in Figure 1. Equipped with this constraint, we develop a clustering approach that

^{*}Département d’informatique de l’ENS, École normale supérieure, CNRS, PSL Research University, 75005 Paris, France.

[†]INRIA

[‡]Czech Institute of Informatics, Robotics and Cybernetics at the Czech Technical University in Prague.

[§]Department of CS & OR (DIRO), Université de Montréal, Montréal.

jointly (i) groups object states with similar appearance and consistent temporal locations with respect to the action and (ii) finds similar manipulation actions separating those object states in the input videos. Our approach exploits the complementarity of both subproblems and finds a joint solution for states and actions. We formulate our problem by adopting a discriminative clustering loss [4] and a joint consistency cost between states and actions. We introduce an effective optimization solution in order to handle the resulting non-convex loss function and the set of spatial-temporal constraints. To evaluate our method, we collect a new video dataset depicting real-life object manipulation actions in realistic videos. Given this dataset for training, our method demonstrates successful discovery of object states and manipulation actions. We also demonstrate that our joint formulation gives an improvement of object state discovery by action recognition and vice versa.

2. Related work

Below we review related work on person-object interaction, recognizing object states, action recognition and discriminative clustering that we employ in our model.

Person-object interactions. Many daily activities involve person-object interactions. Modeling co-occurrences of objects and actions have shown benefits for recognizing actions in [11, 19, 26, 35, 47]. Recent work has also focused on building realistic datasets with people manipulating objects, *e.g.* in instructional videos [3, 32, 37] or while performing daily activities [38]. We build on this work but focus on joint modeling and recognition of actions and *object states*.

States of objects. Prior work has addressed recognition of object attributes [15, 33, 34], which can be seen as different object states in some cases. Differently from our approach, these works typically focus on classifying still images, do not consider human actions and assume an *a priori* known list of possible attributes. Closer to our setting, Isola *et al.* [22] discover object states and transformations between them by analyzing large collections of still images downloaded from the Internet. In contrast, our method does not require annotations of object states. Instead, we use the dynamics of consistent manipulations to discover object states in the video with minimal supervision. In [10], the authors use consistent manipulations to discover task relevant objects. However, they do not consider object states, rely mostly on first person cues (such as gaze) and take advantage of the fact that videos are taken in a single controlled environment.

Action recognition. Most of the prior work on action recognition has focused on designing features to describe time intervals of a video using motion and appearance [30, 39, 43, 44]. This is effective for actions such as *dancing* or *jumping*, however, many of our daily activities are best

distinguishable by their effect on the environment. For example, *opening door* and *closing door* can look very similar using only motion and appearance descriptors but their outcome is completely different. This observation has been used to design action models in [16, 17, 45]. In [45], for example, the authors propose to learn an embedding in which a given action acts as a transformation of features of the video. In our work we localize objects and recognize changes of their states using manipulation actions as a supervisory signal. Related to ours is also the work of Fathi *et al.* [16] who represent actions in egocentric videos by changes of appearance of objects (also called object states), however, their method requires manually annotated precise temporal localization of actions in training videos. In contrast, we focus on (non-egocentric) Internet videos depicting real-life object manipulations where actions are performed by different people in a variety of challenging indoor/outdoor environments. In addition, our model jointly learns to recognize both actions and object states with only minimal supervision.

Discriminative clustering. Our model builds on unsupervised discriminative clustering methods [4, 41, 46] that group data samples according to a simultaneously learned classifier. Such methods can incorporate (weak) supervision that helps to steer the clustering towards a preferred solution [5, 13, 21, 24, 42]. In particular, we build on the discriminative clustering approach of [4] that has been shown to perform well in a variety of computer vision problems [5]. It leads to a quadratic optimization problem where different forms of supervision can be incorporated in the form of (typically) linear constraints. Building on this formalism, we develop a model that jointly finds object states and temporal locations of actions in the video. Part of our object state model is related to [25], while our action model is related to [7]. However, we introduce new spatial-temporal constraints together with a novel joint cost function linking object states and actions, as well as new effective optimization techniques.

Contributions. The contributions of this work are three-fold. First, we develop a new discriminative clustering model that jointly discovers object states and temporally localizes associated manipulation actions in video. Second, we introduce an effective optimization algorithm to handle the resulting non-convex constrained optimization problem. Finally, we experimentally demonstrate that our model discovers key object states and manipulation actions from input videos with minimal supervision.

3. Modeling manipulated objects

We are given a set of N clips that contain a common **manipulation** of the same *object* (such as “**open** an *oyster*”). We also assume that we are given an *a priori* model of the corresponding object in the form of a pre-trained ob-

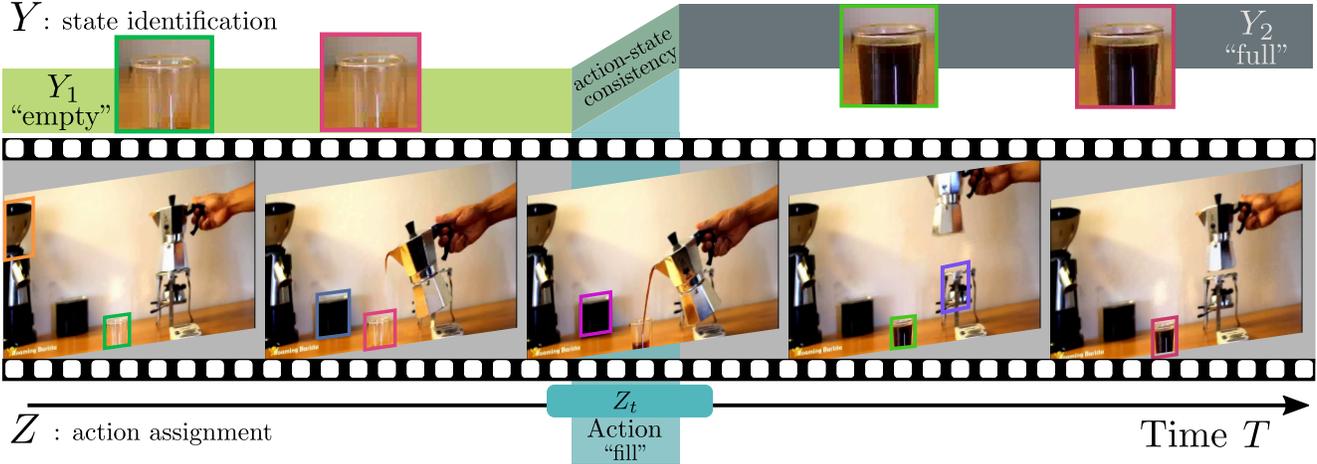


Figure 2: Given a set of clips that depict a manipulated object, we wish to automatically discover the main states that the object can take along with localizing the associated manipulation action. In this example, we show one video of someone filling a coffee cup. The video starts with an empty cup (*state 1*), which is filled with coffee (**action**) to become full (*state 2*). Given imperfect object detectors, we wish to assign to the valid object candidates either the initial state or the final state (encoded in Y). We also want to localize the manipulating action in time (encoded in Z) while maintaining a joint action-state consistency.

ject detector [18]. Given these inputs, our goal is twofold: (i) localize the temporal extent of the action and (ii) spatially/temporally localize the manipulated object and identify its states over time. This is achieved by jointly clustering the appearances of an object (such as an “oyster”) appearing in all clips into two classes, corresponding to the two different states (such as “closed” and “open”), while at the same time temporally localizing a consistent “opening” action that separates the two states consistently in all clips. More formally, we formulate the problem as a minimization of a joint cost function that ties together the action prediction in time, encoded in the assignment variable Z , with the object state discovery in space and time, defined by the assignment variable Y :

$$\begin{aligned}
 & \underset{\substack{Y \in \{0,1\}^{M \times 2} \\ Z \in \{0,1\}^T}}{\text{minimize}} && f(Z) + g(Y) + d(Z, Y) && (1) \\
 & \text{s.t.} && \underbrace{Z \in \mathcal{Z}}_{\substack{\text{saliency of action} \\ \text{Action localization}}} && \text{and} && \underbrace{Y \in \mathcal{Y}}_{\substack{\text{ordering + non overlap} \\ \text{Object state labeling}}}
 \end{aligned}$$

where $f(Z)$ is a discriminative clustering cost to temporally localize the action in each clip, $g(Y)$ is a discriminative clustering cost to identify and localize the different object states and $d(Z, Y)$ is a joint cost that relates object states and actions together. T denotes the total length of all video clips and M denotes the total number of tracked object candidate boxes (tracklets). In addition, we impose constraints \mathcal{Y} and \mathcal{Z} that encode additional structure of the problem: we localize the action with its most salient time interval per clip (“saliency”); we assume that the ordering

of object states is consistent in all clips (“ordering”) and that only one object is manipulated at a time (“non overlap”).

In the following, we proceed with describing different parts of the model (1). In Sec. 3.1 we describe the cost function for the discovery of object states. In Sec. 3.2 we detail our model for action localization. Finally, in Sec. 3.3 we describe and motivate the joint cost d .

3.1. Discovering object states

The goal here is to both (i) spatially localize the manipulated object and (ii) temporally identify its individual states. To address the first goal, we employ pre-trained object detectors. To address the second goal, we formulate the discovery of object states as a discriminative clustering task with constraints. We obtain candidate object detections using standard object detectors pre-trained on large scale existing datasets such as ImageNet [12]. We assume that each clip n is accompanied with a set of M_n tracklets¹ of the object of interest.

We formalize the task of localizing the states of objects as a discriminative clustering problem where the goal is to find an assignment matrix $Y_n \in \{0, 1\}^{M_n \times 2}$, where $(Y_n)_{mk} = 1$ indicates that the m -th tracklet represents the object in state k . We also allow a complete row of Y_n to be zero to encode that no state was assigned to the corresponding tracklet. This is to model the possibility of false

¹In this work, we use short tracks of objects (less than one second) that we call tracklet. We want to avoid long tracks that continue across a state change of objects. By using the finer granularity of tracklets, our model has the ability to correct for detection mistakes within a track as well as identify more precisely the state change.

positive detections of an object, or that another object of the same class appears in the video, but is not manipulated and thus is not undergoing any state change. In detail, we minimize the following discriminative clustering cost [4]:²

$$g(Y) = \min_{W_s \in \mathbb{R}^{d_s \times 2}} \underbrace{\frac{1}{2M} \|Y - X_s W_s\|_F^2}_{\text{Discriminative loss on data}} + \underbrace{\frac{\mu}{2} \|W_s\|_F^2}_{\text{Regularizer}} \quad (2)$$

where W_s is the object state classifier that we seek to learn, μ is a regularization parameter and X_s is a $M \times d_s$ matrix of features, where each row is a d_s -dimensional (state) feature vector storing features for one particular tracklet. The minimization in W_s actually leads to a convex quadratic cost function in Y (see [4]). The first term in (2) is the discriminative loss on the data that measures how easily the input data X_s is classified by the linear classifier W_s when the object state assignment is given by matrix Y . In other words, we wish to find a labeling Y for given object tracklets into two states (or no state) so that their appearance features X are easily classified by a linear classifier. To steer the cost towards the right solution, we employ the following constraints (encoded by $Y \in \mathcal{Y}$ in (1)).

Only one object is manipulated at a time : non overlap constraint. As it is common in instructional videos, we assume that only one object can be manipulated at a given time. However, in practice, it is common to have multiple (spatially diverse) tracklets that occur at the same time, for example, due to a false positive detection in the same frame. To overcome this issue, we impose that at most one tracklet can be labeled as belonging to *state 1* or *state 2* at any given time. We refer to this constraint as “*non overlap*” in problem (1).

state 1 \rightarrow Action \rightarrow state 2: ordering constraints. We assume that the manipulating action transforms the object from an initial state to a final state and that both states are present in each video. This naturally introduces two constraints. The first one is the ordering constraints on the labeling Y_n , *i.e.* the *state 1* should occur before *state 2* in each video. The second constraint imposes that we have *at least one* tracklet labeled as *state 1* and at least one tracklet labeled as *state 2*. We call this last constraint the “**at least one**” constraint in contrast to forcing “*exactly one*” ordered prediction as previously proposed in a discriminative clustering approach on video for action localization [7]. This new type of constraint brings additional optimization challenges that we address in Section 4.2.

3.2. Action localization

Our action model is equivalent to the one of [7] applied to only *one action*. More precisely, the goal is to find an assignment matrix $Z_n \in \{0, 1\}^{T_n}$ for each clip n , where

²We concatenate all the variables Y_n into one $M \times 2$ matrix Y .

$Z_{nt} = 1$ encodes that the t -th time interval of video is assigned to an action and $Z_{nt} = 0$ encodes that no action is detected in interval t . The cost that we minimize for this problem is similar to the object states cost:

$$f(Z) = \min_{W_v \in \mathbb{R}^{d_v}} \underbrace{\frac{1}{2T} \|Z - X_v W_v\|_F^2}_{\text{Discriminative loss on data}} + \underbrace{\frac{\lambda}{2} \|W_v\|_F^2}_{\text{Regularizer}}, \quad (3)$$

where W_v is the action classifier, λ is a regularization parameter and X_v is a matrix of visual features. We constrain our model to predict *exactly* one time interval for an action per clip, an approach for actions that was shown to be beneficial in a weakly supervised setting [7] (referred to as “*action saliency*” constraint). As will be shown in experiments, this model *alone* is incomplete because the clips in our dataset can contain other actions that do not manipulate the object of interest. Our central contribution is to propose a *joint formulation* that links this action model with the object state prediction model, thereby resolving the ambiguity of actions. We detail the joint model next.

3.3. Linking actions and object states

Actions in our model are directly related to changes in object states. We therefore want to enforce consistency between the two problems. To do so, we design a novel joint cost function that operates on the action video labeling Z_n and the state tracklet assignment Y_n for each clip. We want to impose a constraint that the action occurs in between the presence of the two different object states. In other words, we want to penalize the fact that state 1 is detected *after* the action happens, or the fact that state 2 is triggered *before* the action occurs.

Joint cost definition. We propose the following joint symmetric cost function for each clip:

$$d(Z_n, Y_n) = \sum_{y \in \mathcal{S}_1(Y_n)} [t_y - t_{Z_n}]_+ + \sum_{y \in \mathcal{S}_2(Y_n)} [t_{Z_n} - t_y]_+, \quad (4)$$

where t_{Z_n} and t_y are the times when the action Z_n and the tracklet y occur in a clip n , respectively. $\mathcal{S}_1(Y_n)$ and $\mathcal{S}_2(Y_n)$ are the tracklets in the n -th clip that have been assigned to state 1 and state 2, respectively. Finally $[x]_+$ is the positive part of x . In other words, the function penalizes the inconsistent assignment of objects states Y_n by the amount of time that separates the incorrectly assigned tracklet and the manipulation action in the clip. The overall joint cost is the sum over all clips weighted by a scaling hyperparameter $\nu > 0$:

$$d(Z, Y) = \nu \frac{1}{T} \sum_{n=1}^N d(Z_n, Y_n). \quad (5)$$

4. Optimization

Optimizing problem (1) poses several challenges that need to be addressed. First, we propose a relaxation of the

integer constraints and the distortion function (Section 4.1). Second, we optimize this relaxation using Frank-Wolfe with a new dynamic program able to handle our tracklet constraints (Section 4.2). Finally, we introduce a new rounding technique to obtain an integer candidate solution to our problem (Section 4.3).

4.1. Relaxation

Problem (1) is NP-hard in general [31] due to its specific integer constraints. Inspired by the approach of [6] that was successful to approximate combinatorial optimization problems, we propose to use the tightest convex relaxation of the feasible subset of binary matrices by taking its convex hull. As our variables now can take values in $[0, 1]$, we also have to propose a consistent extension for the different cost functions to handle fractional values as input. For the cost functions f and g , we can directly take their expression on the relaxed set as they are already expressed as (convex) quadratic functions. Similarly, for the joint cost function d in (4), we use its natural bilinear relaxation:

$$d(Z_n, Y_n) = \sum_{i=1}^{M_n} \sum_{t=1}^{T_n} \left((Y_n)_{i1} Z_{nt} [t_{ni} - t]_+ + (Y_n)_{i2} Z_{nt} [t - t_{ni}]_+ \right), \quad (6)$$

where t_{ni} denotes the video time of tracklet i in clip n . This relaxation is equal to the function (4) on the integer points. However, it is not jointly convex in Y and Z , thus we have to design an appropriate optimization technique to obtain good (relaxed) candidate solutions, as described next.

4.2. Joint optimization using Frank-Wolfe

When dealing with a constrained optimization problem for which it is easy to solve linear programs but difficult to project on the feasible set, the Frank-Wolfe algorithm is an excellent choice [23, 29]. It is exactly the case for our relaxed problem, where the linear program over the convex hull of feasible integer matrices can be solved efficiently via dynamic programming. Moreover, [28] recently showed that the Frank-Wolfe algorithm with line-search converges to a stationary point for non-convex objectives at a rate of $O(1/\sqrt{k})$. We thus use this algorithm for the joint optimization of (1). As the objective is quadratic, we can perform exact line-search analytically, which speeds up convergence in practice. Finally, in order to get a good initialization for both variables Z and Y , we first optimize separately $f(Z)$ and $g(Y)$ (without the non-convex $d(Z, Y)$), which are both convex functions.

Dynamic program for the tracklets. In order to apply the Frank-Wolfe algorithm, we need to solve a linear program (LP) over our set of constraints. Previous work has explored “*exact one*” ordering constraints for time localization problems [6]. Differently here, we have to deal with

the spatial (non overlap) constraint *and* finding “*at least one*” candidate tracklet per state. To deal with these two requirements, we propose a novel dynamic programming approach. First, the “at least one” constraint is encoded by having a memory variable which indicates whether state 1 or state 2 have already been visited. This variable is used to propose valid state decisions for consecutive tracklets. Second, the challenging “non-overlap” tracklet constraint is included by constructing valid left-to-right paths in a cost matrix while carefully considering the possible authorized transitions. We provide details of the formulation in the appendix [2]. In addition, we show in section 5.2 that these new constraints are *key* for the success of the method.

4.3. Joint rounding method

Once we obtain a candidate solution of the relaxed problem, we have to round it to an integer solution in order to make predictions. Previous works [3, 7] have observed that using the learned W^* classifier for rounding gave better results than other possible alternatives. We extend this approach to our joint setup by proposing the following new rounding procedure. We optimize problem (1) but fix the values of W in the discriminative clustering costs. Specifically, we minimize the following cost function over the integer points $Z \in \mathcal{Z}$ and $Y \in \mathcal{Y}$:

$$\frac{1}{2T} \|Z - X_v W_v^*\|_F^2 + \frac{1}{2M} \|Y - X_s W_s^*\|_F^2 + d(Z, Y), \quad (7)$$

where W_v^* and W_s^* are the classifier weights obtained at the end of the relaxed optimization. Because $y^2 = y$ when y is binary, (7) is actually a *linear* objective over the binary matrix Y_n for Z_n fixed. Thus we can optimize (7) *exactly* by solving a dynamic program on Y_n for each of the T_n possibilities of Z_n , yielding $O(M_n T_n)$ time complexity per clip (see the appendix [2] for details).

5. Experiments

In this section, we first describe our dataset, the object tracking pipeline and the feature representation for object tracklets and videos (Section 5.1). We consider two experimental set-ups. In the first weakly-supervised set-up (Section 5.2), we apply our method on a set of video clips which we know contain the action of interest but do not know its precise temporal localization. In the second, more challenging “in the wild” set-up (Section 5.3), the input set of weakly-supervised clips is obtained by automatic processing of text associated with the videos and hence may contain erroneous clips that do not contain the manipulation action of interest. The data and code are available online [1].

5.1. Dataset and features

Dataset of manipulation actions. We build a dataset of manipulation actions by collecting videos from different

sources: the instructional video dataset introduced in [3], the Charades dataset from [38], and some additional videos downloaded from YouTube. We focus on “third person” videos (rather than egocentric) as such videos depict a variety of people in different settings and can be obtained on a large scale from YouTube. We annotate the precise temporal extent of seven different actions³ applied to five distinct objects⁴. This results in 630 annotated occurrences of ground truth manipulation action.

To evaluate object state recognition, we define a list of two states for each object. We then run automatic object detector for each involved object, track the detected object occurrences throughout the video and then subdivide the resulting long tracks into short tracklets. Finally, we label ground truth object states for tracklets within ± 40 seconds of each manipulation action. We label four possible states: *state 1*, *state 2*, *ambiguous state* or *false positive detection*. The ambiguous state covers the (not so common) in-between cases, such as cup half-full. In total, we have 19,499 fully annotated tracklets out of which: 35% cover *state 1* or *state 2*, 25% are ambiguous, and 40% are false positives. Note that this annotation is only used for evaluation purpose, and not by any of our models. Detailed statistics of the dataset are given in the appendix [2].

Object detection and tracking. In order to obtain detectors for the five objects, we finetune the FastRCNN network [18] with training data from ImageNet [12]. We use bounding box annotations from ImageNet when available (e.g. the “wheel” class). For the other classes, we manually labeled more than 500 instances per class. In our set-up with only moderate amount of training data, we observed that class-agnostic object proposals combined with FastRCNN performed better than FasterRCNN [36]. In detail, we use geodesic object proposals [27] and set a relatively low object detection threshold (0.4) to have good recall. We track objects using a generic KLT tracker from [5]. The tracks are then post-processed into shorter tracklets that last about one second and thus are likely to have only one object state.

Object tracklet representation. For each detected object, represented by a set of bounding boxes over the course of the tracklet, we compute a CNN feature from each (extended) bounding box that we then average over the length of the tracklet to get the final representation. The CNN feature is extracted with a ROI pooling [36] of ResNet50 [20]. The ROI pooling notably allows to capture some context around the object which is important for some cases (e.g. *wheel* “on” or “off” the car). The resulting feature descriptor of each object tracklet is 8,192 dimensional.

Representing video for recognizing actions. Following

³put the wheel on the car (47 clips), withdraw the wheel from the car (46), place a plant inside a pot (27), open an oyster (28), open a refrigerator (234), close a refrigerator (191) and pour coffee (57).

⁴car wheel, flower pot, oyster, refrigerator and coffee cup.

the approach of [3, 6, 7], each video is divided into chunks of 10 frames that are represented by a motion and appearance descriptor averaged over 30 frames. For the motion we use a 2,000 dimensional bag-of-word representation of histogram of local optical flow (HOF) obtained from Improved Dense Trajectories [44]. Following [3], we add an appearance vector that is obtained from a 1,000 dimensional bag-of-word vector of conv5 features from VGG16 [40]. This results in a 3,000 dimensional feature vector for each chunk of 10 frames.

5.2. Weakly supervised object state discovery

Experimental setup. We first apply our method in a weakly supervised set-up where for each action we provide an input set of clips, where we know the action occurs somewhere in the clip but we do not provide the precise temporal localization. Each clip may contain other actions that affect other objects or actions that do not affect any object at all (e.g. walking / jumping). The input clips are about 20s long and are obtained by taking approximately ± 10 s of each annotated manipulation action.

Evaluation metric: average precision. For all variants of our method, we use the rounded solution that reached the smallest objective during optimization. We evaluate these predictions with a precision score averaged over all the videos. A temporal action localization is said to be correct if it falls within the ground truth time interval. Similarly, a state prediction is correct if it matches the ground truth state.⁵ Note that a “precision” metric is reasonable in our set-up as our method is forced to predict in all videos, i.e. the recall level is fixed to all videos and the method cannot produce high precision with low recall.

Hyperparameters. In all methods that involve a discriminative clustering objective, we used $\lambda = 10^{-2}$ (action localization) and $\mu = 10^{-4}$ (state discovery) for all 7 actions. For joint methods that optimize (1), we set the weight ν of the distortion measure (5) to 1.

State discovery results. Results are shown in the top part of Table 1. In the following, we refer to “State only” whenever we use our method without looking at the action cost or the distortion measure (1). We compare to two baselines for the state discovery task. Baseline (a) evaluates chance performance. Baseline (b) performs K-means clustering of the tracklets with $K = 3$ (2 clusters for the states and 1 for false positives). We report performance of the best assignment for the solution with the lowest objective after 10 different initializations. Baseline (c) is obtained by running our “State only” method while using random features for tracklet representation as well as “at least one ordering” and “non overlap” constraints. We use random features to avoid non-trivial analytic derivation for the “Constraints only” performance. This baseline reveals the difficulty of the problem

⁵In particular, we count “ambiguous” labels as incorrect.

Method		put wheel	remove wheel	fill pot	open oyster	fill coff.cup	open fridge	close fridge	Average
State discovery	(a) Chance	0.10	0.11	0.10	0.07	0.06	0.10	0.10	0.09
	(b) Kmeans	0.25	0.12	0.11	0.23	0.14	0.19	0.22	0.18
	(c) Constraints only	0.35	0.38	0.35	0.36	0.31	0.29	0.42	0.35
	(d) Salient state only	0.35	0.48	0.35	0.38	0.30	0.40	0.37	0.38
	(e) At least one state only	0.43	0.55	0.46	0.52	0.29	0.43	0.39	0.44
	(f) Joint model	0.52	0.59	0.50	0.45	0.39	0.47	0.47	0.48
	(g) Joint model + det. scores.	0.47	0.65	0.50	0.61	0.44	0.46	0.43	0.51
	(h) Joint + GT act. feat.	0.55	0.56	0.56	0.52	0.46	0.45	0.49	0.51
Action localization	(i) Chance	0.31	0.20	0.15	0.11	0.40	0.23	0.17	0.22
	(ii) [7]	0.24	0.13	0.11	0.14	0.26	0.29	0.23	0.20
	(iii) [7] + object cues	0.24	0.13	0.26	0.07	0.84	0.33	0.37	0.32
	(iv) Joint model	0.67	0.57	0.48	0.32	0.82	0.57	0.44	0.55
	(v) Joint + GT stat. feat.	0.72	0.66	0.44	0.46	0.86	0.55	0.44	0.59

Table 1: State discovery (top) and action localization results (bottom).

and quantifies improvement brought by the ordering constraints. The next two methods are “State only” variants. Method (d) corresponds to a replacement of the “at least one constraint” by an “exactly one constraint” while the method (e) uses our new constraint. Finally, we report three joint methods that use our new joint rounding technique (7) for prediction. Method (f) corresponds to our joint method that optimizes (1). Method (g) is a simple improvement taking into account object detection score in the objective (details below). Finally, method (h) is our joint method but using the action ground truth labels as video features in order to test the effect of having perfect action localization for the task of object state discovery.

We first note that method (e) outperforms (d), thus highlighting the importance of the “at least one” constraint for modeling object states. While the saliency approach (taking only the most confident detection per video) was useful for action modeling in [7], it is less suitable for our set-up where multiple tracklets can be in the same state. The joint approach with actions (f) outperforms the “State only” method (e) on 6 out of 7 actions and obtains better average performance, confirming the benefits of joint modeling of actions and object states. Using ground truth action locations further improves results (cf. (h) against (f)). Our weakly supervised approach (f) performs not much lower compared to using ground truth actions (h), except for the states of the coffee cup (empty/full). In this case we observe that a high number of false positive detections confuses our method. A simple way to address this issue is to add the object detection score into the objective of our method, which then prefers to assign object states to higher scoring object candidates further reducing the effect of false positives. This can be done easily by adding a linear cost reflecting the object detection score to objective (1). We denote this modified method “(g) Joint model + det. scores”. This method achieves the best average performance and highlights that additional information can be easily added to our model.

Action localization results. We compare our method to three different baselines and give results in the bottom part of Table 1. Baseline (i) corresponds to chance performance,

where the precision for each clip is simply the proportion of the entire clip taken by the ground truth time interval. Baseline (ii) is the method introduced in [7] used here with only one action. It also corresponds to a special case of our method where the object state part of the objective in equation (1) is turned off (salient action only). Interestingly, this baseline is actually worse than chance for several actions. This is because without additional information about objects, this method localizes other common actions in the clip and not the action manipulating the object of interest. This also demonstrates the difficulty of our experimental set-up where the input video clips often contain multiple different actions. To address this issue, we also evaluate baseline (iii), which complements [7] with the additional constraint that the action prediction has to be within the first and the last frame where the object of interest is detected, improving the overall performance above chance. Our joint approach (iv) consistently outperforms these baselines on all actions, thus showing again the strong link between object states and actions. Finally, the approach (v) is the analog of method (g) for action localization where we use ground truth state labels as tracklet features in our joint formulation showing that the action localization can be further improved with better object state descriptors. In addition, we also compare to a supervised baseline. The average obtained performance is 0.58 which is not far from our method. This demonstrates the potential of using object states for action localization. More details on this experiment are provided in the appendix [2].

Benefits of joint object-action modeling. We observe that the joint modeling of object states and actions benefits both tasks. This effect is even stronger for actions. Intuitively, knowing perfectly the object states reduces a lot the search space for action localization. Moreover, despite the recent major progress in object recognition using CNNs, action recognition still remains a hard problem with much room for improvement. Qualitative results are shown in Fig. 3 and failure cases of our method are discussed in [2].

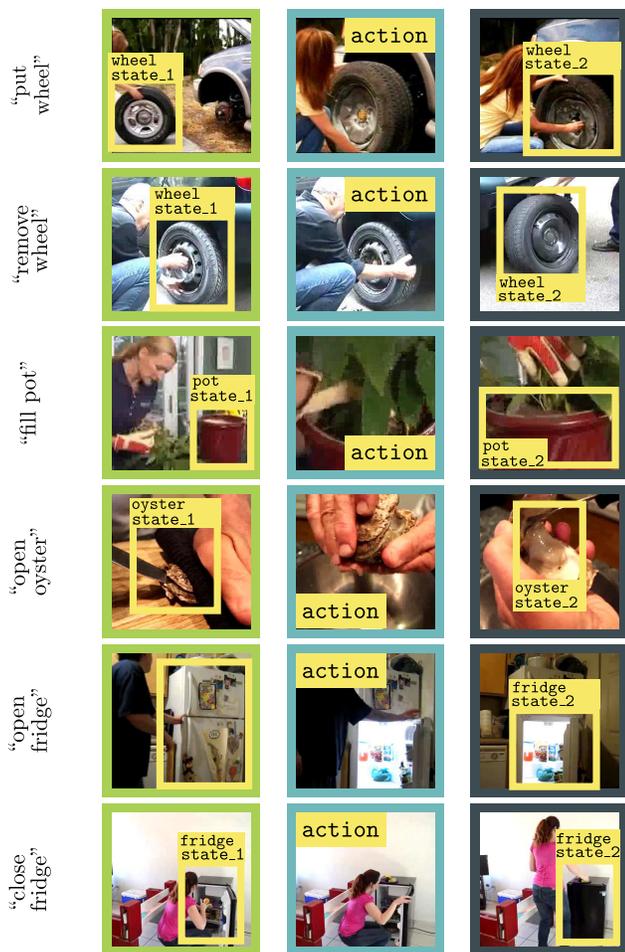


Figure 3: Qualitative results for joint action localization (middle) and state discovery (left and right) (see Fig. 1 for “fill coffee cup”).

5.3. Object state discovery in the wild

Towards the discovery of a large number of manipulation actions and state changes, we next apply our method in an automatic setting, where action clips have been obtained using automatic text-based retrieval.

Clip retrieval by text. Instructional videos [3, 32, 37] usually come with a narration provided by the speaker describing the performed sequence of actions. In this experiment, we keep only such *narrated instructional videos* from our dataset. This results in the total of 140 videos that are 3 minutes long in average. We extract the narration in the form of subtitles associated with the video. These subtitles have been directly downloaded from YouTube and have been obtained either by Youtube’s Automatic Speech Recognition (ASR) or provided by the users.

We use the resulting text to retrieve clip candidates that may contain the action modifying the state of an object. Obtaining the approximate temporal location of actions from the transcribed narration is still very challenging due to ambiguities in language (“undo bolt” and “loosen nut” refer to

	Method	put wheel	remove wheel	fill pot	open oyster	fill coff.cup	Ave.
State disc.	(c) Cstrs only	0.23	0.34	0.25	0.29	0.11	0.24
	State + det. sc.	0.33	0.48	0.28	0.40	0.13	0.32
	(g) Joint	0.38	0.53	0.25	0.43	0.20	0.36
	(g) Curated	0.63	0.68	0.63	0.63	0.53	0.62
Action local.	(i) Chance	0.14	0.10	0.06	0.10	0.15	0.11
	(iii) Action	0.05	0.10	0.00	0.15	0.25	0.11
	(iv) Joint	0.30	0.30	0.20	0.20	0.20	0.24
	(iv) Curated	0.53	0.35	0.32	0.40	0.59	0.44

Table 2: Results on noisy clips automatically retrieved by text.

the same manipulation) and only coarse temporal localization of the action provided by the narration. Given a manipulation action such as “remove tire”, we first find positive and negative sentences relevant for the action from an instruction website such as Wikihow. We then train a linear SVM classifier [9] on bigram text features. Finally, we use the learned classifier to score clips from the input instructional videos. In detail, the classifier is applied in a sliding window of 10 words finding the best scoring window in each input video. The clip candidates are then obtained by trimming the input videos 5 seconds before and 15 seconds after the timing of the best scoring text window to account for the fact that people usually perform the action after having talked about it. We apply our method on the top 20 video clips based on the SVM score for each manipulation action. More details about this process are provided in the appendix [2].

Results. As shown in Table 2, the pattern of results, where our joint method performs the best, is similar to the weakly supervised set-up described in Sec. 5.2. This highlights the robustness of our model to noisy input data – an important property for scaling-up the method to Internet scale datasets. To assess how well our joint method could do with perfect retrieval, we also report results for a “Curated” set-up where we replace the automatically retrieved clips with the 20s clips used in Sec. 5.2 for the corresponding videos.

6. Conclusion and future work

We have described a joint model that relates object states and manipulation actions. Given a set of input videos, our model both localizes the manipulation actions *and* discovers the corresponding object states. We have demonstrated that our joint approach improves performance of both object state recognition and action recognition. More generally, our work provides evidence that actions should be modeled in the larger context of goals and effects. Finally, our work opens up the possibility of Internet-scale learning of manipulation actions from narrated video sequences.

Acknowledgments This research was supported in part by a Google Research Award, ERC grants Activia (no. 307574) and LEAP (no. 336845), the CIFAR Learning in Machines & Brains program and ESIF, OP Research, development and education Project IMPACT No. CZ.02.1.01/0.0/0.0/15 003/0000468.

References

- [1] Project webpage (code/dataset). <http://www.di.ens.fr/willow/research/objectstates/>. 5
- [2] Supplementary material (appendix) for the paper. <http://arxiv.org/abs/1702.02738>. 5, 6, 7, 8
- [3] J.-B. Alayrac, P. Bojanowski, N. Agrawal, I. Laptev, J. Sivic, and S. Lacoste Julien. Unsupervised learning from narrated instruction videos. In *CVPR*, 2016. 2, 5, 6, 8
- [4] F. Bach and Z. Harchaoui. DIFFRAC: A discriminative and flexible framework for clustering. In *NIPS*, 2007. 2, 4
- [5] P. Bojanowski, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Finding actors and actions in movies. In *ICCV*, 2013. 2, 6
- [6] P. Bojanowski, R. Lajugie, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Weakly supervised action labeling in videos under ordering constraints. In *ECCV*, 2014. 5, 6
- [7] P. Bojanowski, R. Lajugie, E. Grave, F. Bach, I. Laptev, J. Ponce, and C. Schmid. Weakly-supervised alignment of video with text. In *ICCV*, 2015. 2, 4, 5, 6, 7
- [8] 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
- [9] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, pages 273–297, 1995. 8
- [10] D. Damen, T. Leelasawassuk, O. Haines, A. Calway, and W. Mayol-Cuevas. You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video. In *BMVA*, 2014. 2
- [11] V. Delaitre, J. Sivic, and I. Laptev. Learning person-object interactions for action recognition in still images. In *NIPS*, 2011. 2
- [12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 3, 6
- [13] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros. What makes Paris look like Paris? *SIGGRAPH*, 2012. 2
- [14] K. Duan, D. Parikh, D. Crandall, and K. Grauman. Discovering localized attributes for fine-grained recognition. In *CVPR*, pages 3474–3481, 2012. 1
- [15] A. Farhadi, I. E. Lim, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 1, 2
- [16] A. Fathi and J. M. Rehg. Modeling actions through state changes. In *CVPR*, 2013. 2
- [17] B. Fernando, E. Gavves, M. J. Oramas, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, 2015. 2
- [18] R. Girshick. Fast R-CNN. In *ICCV*, 2015. 3, 6
- [19] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *PAMI*, 31(10):1775–1789, 2009. 2
- [20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6
- [21] D.-A. Huang, L. Fei-Fei, and J. C. Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *ECCV*, 2016. 2
- [22] P. Isola, J. J. Lim, and E. H. Adelson. Discovering states and transformations in image collections. In *CVPR*, 2015. 1, 2
- [23] M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML*, 2013. 5
- [24] A. Jain, A. Gupta, M. Rodriguez, and L. Davis. Representing videos using mid-level discriminative patches. In *CVPR*, pages 2571–2578, 2013. 2
- [25] A. Joulin, K. Tang, and L. Fei-Fei. Efficient image and video co-localization with Frank-Wolfe algorithm. In *ECCV*, 2014. 2
- [26] H. Kjellström, J. Romero, and D. Kragić. Visual object-action recognition: Inferring object affordances from human demonstration. *CVIU*, 115(1):81–90, 2011. 2
- [27] P. Krhenbhl and V. Koltun. Geodesic object proposals. In *ECCV*, 2014. 6
- [28] S. Lacoste-Julien. Convergence rate of Frank-Wolfe for non-convex objectives. *arXiv:1607.00345*, 2016. 5
- [29] S. Lacoste-Julien and M. Jaggi. On the global linear convergence of Frank-Wolfe optimization variants. In *NIPS*, 2015. 5
- [30] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 2
- [31] E. M. Loiola, N. M. M. de Abreu, P. O. Boaventura-Netto, P. Hahn, and T. Querido. A survey for the quadratic assignment problem. *EJOR*, 176(2):657–690, 2007. 5
- [32] J. Malmaud, J. Huang, V. Rathod, N. Johnston, A. Rabinovich, and K. Murphy. What’s cookin’? Interpreting cooking videos using text, speech and vision. In *NAACL*, 2015. 2, 8
- [33] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011. 2
- [34] G. Patterson, C. Xu, H. Su, and J. Hays. The SUN attribute database: Beyond categories for deeper scene understanding. *IJCV*, 2014. 2
- [35] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *CVPR*, 2012. 2
- [36] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 6
- [37] O. Sener, A. Zamir, S. Savarese, and A. Saxena. Unsupervised semantic parsing of video collections. In *ICCV*, 2015. 2, 8
- [38] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 2, 6
- [39] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 2
- [40] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2014. 6
- [41] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012. 2
- [42] K. Tang, A. Joulin, L.-J. Li, and L. Fei-Fei. Co-localization in real-world images. In *CVPR*, 2014. 2
- [43] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*, 2015. 2

- [44] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. [2](#), [6](#)
- [45] X. Wang, A. Farhadi, and A. Gupta. Actions ~ transformations. In *CVPR*, 2016. [2](#)
- [46] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. In *NIPS*, 2004. [2](#)
- [47] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *ICCV*, 2011. [2](#)