Sentence Directed Video Object Codiscovery

Haonan Yu[1] · **Jeffrey Mark Siskind**[2]

Received: 2 July 2016 / Accepted: 17 May 2017 / Published online: 9 June 2017

Abstract Video object codiscovery can leverage the weak semantic constraint implied by sentences that describe the video content. Our codiscovery method, like other object codetection techniques, does not employ any pretrained object models or detectors. Unlike most prior work that focuses on codetecting large objects which are usually salient both in size and appearance, our method can discover small or medium sized objects as well as ones that may be occluded for part of the video. More importantly, our method can codiscover multiple object instances of different classes within a single video clip. Although the semantic information employed is usually simple and weak, it can greatly boost performance by constraining the hypothesized object locations. Experiments show promising results on three datasets: an average IoU score of 0.423 on a new dataset with 15 object classes, an average IoU score of 0.373 on a subset of CAD-120 with 5 object classes, and an average IoU score of 0.358 on a subset of MPII-Cooking with 7 object classes. Our result on this subset of MPII-Cooking improves upon those of the previous state-of-the-art methods by significant margins.

Keywords Video · Object codiscovery · Sentences

1 Introduction

We address the problem of video object codiscovery: naming and localizing novel objects in a set of videos, by placing bounding boxes around those objects, *without* any pretrained object detectors. This problem is essentially one of video object codetection: given a set of videos that contain instances of a common object class, locate those instances simultaneously. However, our work differs from most prior codetection work in two crucial ways. First, our method can codetect small or medium sized objects, as well as ones that are occluded for part of the video. Second, it can codetect multiple object instances of different classes both within a single video clip and across a set of video clips. Thus, following Srikantha and Gall (2014), we call our task *codiscovery*, to distinguish it from approaches that are only able to codetect a single large salient object per image or video as well as to distinguish it from approaches that are only able to codetect a single object class across a codetection set. Our approach is a form of weakly supervised learning, where hidden structure is inferred from weakly labeled data.

Object codetection with a single class is typically approached by selecting one out of many object proposals per image or frame that maximizes a combination of the confidence scores associated with the selected proposals and the similarity scores between proposal pairs (Tang et al. 2014).

Communicated by Deva Ramanan.

This research was sponsored, in part, by the Army Research Laboratory, accomplished under Cooperative Agreement Number W911NF-10-2-0060, and by the National Science Foundation under Grant No. 1522954-IIS. The views, opinions, findings, conclusions, and recommendations contained in this document are those of the authors and should not be interpreted as representing the official policies, either express or implied, of the Army Research Laboratory, the National Science Foundation, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein.

✉ Haonan Yu
haonanu@gmail.com

Jeffrey Mark Siskind
qobi@purdue.edu

[1] Baidu Research - Institute of Deep Learning, Sunnyvale, CA 94089, USA

[2] School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA

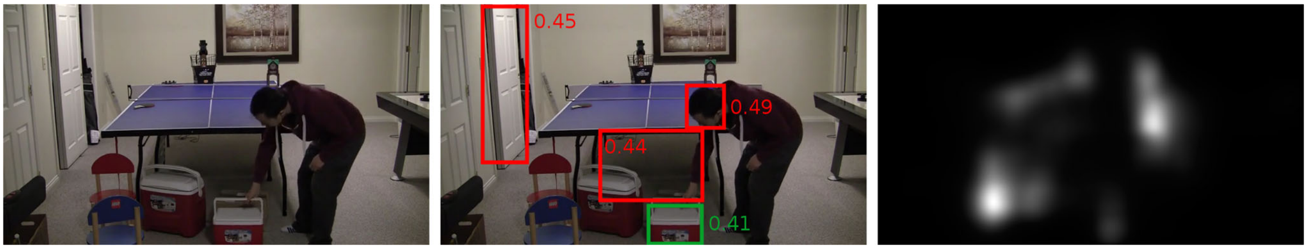


Fig. 1 Object proposal confidence and saliency scores for a sample frame from our new dataset. *Left* the original input video frame. *Middle* several proposals and associated confidence scores produced by the method of Arbelaez et al. (2014). Note that the *red* boxes, which do not correspond to objects, let alone salient ones, all have higher scores than the *green* box, which does denote a salient object. *Right* the saliency map output by the saliency detection method of Jiang et al. (2015), currently the highest ranking method on the MIT saliency benchmark (Bylinskii et al. 2012). Note that the *cooler* is not highlighted as salient. Using these scores as part of the scoring function can drive the codetection process to produce undesired results (Color figure online)

The confidence score indicates how likely an image region corresponds to an object, while the similarity score biases the process to codetect instances with similar appearance. Since codetection is a process of ambiguity resolution, it is generally more difficult to codetect objects from fewer videos than from more.

The confidence score of a proposal can sometimes be a poor indicator of whether a proposal denotes an object, especially when objects are occluded, the lighting is poor, or motion blur exists (e.g., Fig. 1). Salient objects can have low confidence score while nonsalient objects or image regions that do not correspond to objects can have high confidence score. Thus our scoring function does not use the confidence scores produced by the proposal generation mechanism. Instead, we avail ourselves of a different source of constraint on the codiscovery problem. In videos depicting human interaction with objects to be codiscovered, descriptions of such activity can impart *weak* spatial or motion constraint either on a single object or among multiple objects of interest. For example, if the video depicts a “pick up” event, some object should have an upward displacement during this process, which should be detectable even if it is small. This motion constraint will reliably differentiate the object which is being picked up from other stationary background objects. It is weak because it might not totally resolve the ambiguity; other image regions might satisfy this constraint, perhaps due to noise. Similarly, if we know that object A is to the left of object B, then the detection search for object A will weakly affect the detection search for object B, and vice versa. To this end, we extract spatio-temporal constraints from *sentences* that describe the videos and then impose these constraints on the codiscovery process to find the collections of objects that best satisfy these constraints and that are similar within each object class. Even though the constraints implied by a single sentence are usually weak, when accumulated across a set of videos and sentences, they together will greatly prune the detection search space. We call this process *sentence directed* video object codiscovery. It can be viewed as the *inverse* of video captioning/description (Barbu et al. 2012; Das et al. 2013; Guadarrama et al. 2013; Rohrbach et al. 2014; Venugopalan et al. 2015; Yu et al. 2015, 2016) where object evidence (in the form of detections or other visual features) is first produced by pretrained detectors and then sentences are generated given the object appearance and movement.

Our sentence directed codiscovery process produces instances of multiple object classes at a time by its very nature. The sentence we use to describe a video usually contains multiple nouns referring to multiple object instances of different classes. The sentence semantics captures the spatio-temporal relationships between these objects. As a result, the codiscovery of one object class affects that of the others and vice versa. In contrast, as we will see in Sect. 2, all prior codetection methods, whether for images or video, codetect only one common object class at a time: different object classes are codetected independently. Each time they output a single object detection of the same class for each video clip. To the best of our knowledge, our work is the first to codiscover multiple object classes, both in one video clip and across multiple video clips, simultaneously (Fig. 2).

Generally speaking, we extract a set of predicates from each sentence and formulate each predicate around a set of *primitive functions*. The predicates may be verbs (e.g., CARRIED and ROTATED), spatial-relation prepositions (e.g., LEFTOF and ABOVE), motion prepositions (e.g., AWAYFROM and TOWARDS), or adverbs (e.g., QUICKLY and SLOWLY). The sentential predicates are applied to the candidate object proposals as arguments, allowing an overall predicate score to be computed that indicates how well these candidate object proposals satisfy the sentence semantics. We add this predicate score into the codiscovery framework, on top of the original similarity score, to guide the optimization. To the best of our knowledge, this is the first work that uses sentences to guide generic video object codiscovery. To summarize, our approach differs from prior codetection work (Sect. 2) in the following ways:

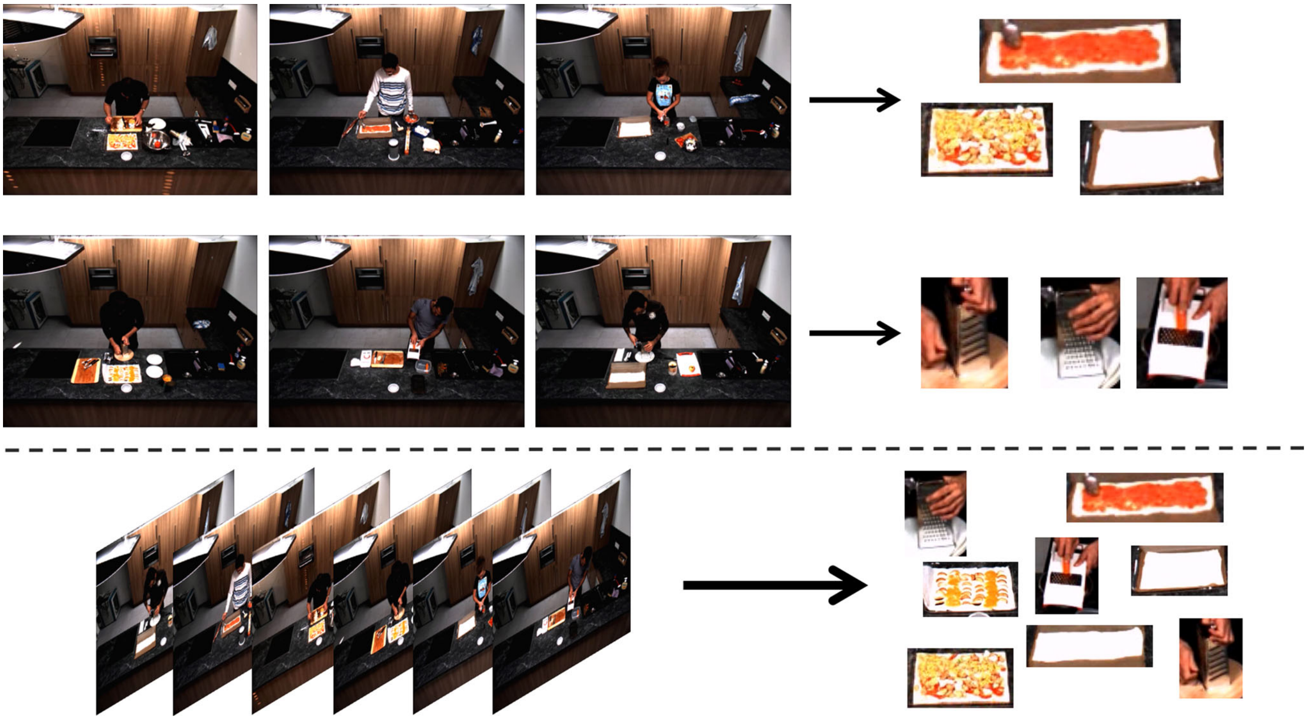


Fig. 2 *Top*: Single-class codetection performed by prior work. Given an object class, a subset of videos is defined for that class and codetection is performed on that class. Object instances of two different classes (e.g., *bread* and *grater*) are codetected independently. *Bottom*: Simultaneous multi-class codiscovery performed by our approach. The two subsets of videos can be pooled together and the object instances of the two classes can be codiscovered simultaneously. Note that such pooling allows two additional *bread* instances in the *grater* video subset to also be codiscovered due to their spatial relationship with the *grater*, even though they are not involved in the human activity of *seasoning bread* that is used by Srikantha and Gall (2014) to codiscover *bread*

(a) Our method can codiscover small or medium sized non-salient objects which can be located anywhere in the field of view.
(b) Our method can codiscover multiple objects simultaneously, both multiple instances and classes within a single video and multiple classes across a set of videos. These objects can be either moving in the foreground or stationary in the background.
(c) Our method leverages sentence semantics to help codiscovery.

We evaluated our approach on three different datasets. The first is a new dataset that contains 15 distinct object classes and 150 video clips with a total of 12,509 frames. The second is a subset of CAD-120 (Koppula et al. 2013), a dataset originally intended for activity recognition. We constructed the subset by selecting videos that contain multiple object instances of interest, to demonstrate our capability of simultaneous multi-class object codiscovery. The subset contains 5 distinct object classes and 75 video clips with a total of 8854 frames. The last is a subset of MPII-Cooking (Rohrbach et al. 2012), the most challenging dataset used in Srikantha and Gall (2014, 2017). The subset contains 7 distinct object classes and 233 video clips with a total of 70,259 frames. On this dataset, we conducted an apples-to-apples comparison of our method with Prest et al. (2012) and Srikantha and Gall (2014, 2017), and show that our method outperforms these prior methods by significant margins. Our approach achieves an average IoU (Intersection-over-Union) score of 0.423 on the new dataset, 0.373 on the subset of CAD-120, and 0.358 on the subset of MPII-Cooking.

Our focus here is the general idea of using *semantic* information of the activity in a video to assist codiscovery. Our contribution is not the particular predicates and parsing rules which we will later use to instantiate this idea. While conceptually, such semantic information is typically conveyed by sentences, our focus is on the computer vision task of *using* such information when processing video, not the natural language processing task of *extracting* such information from sentences. Thus we neither implement nor describe a semantic parser that would perform such extraction. Construction of semantic parsers, either by hand or by using machine learning techniques, is well studied in the field of natural-language processing (Wong and Mooney 2007; Clarke et al. 2010). Indeed, deep learning techniques, such as Recurrent Neural Networks (Elman 1990; Hochreiter and Schmidhuber 1997) and Neural Turing Machines (Graves et al. 2014), are promising candidates for approaching this task. In principle, our codiscovery methods could be coupled with one of a variety of different methods for learning a semantic parser to build a more comprehensive system that learns both to detect objects and process language. However, our goal here is not to build such a comprehensive system and thus we leave the training of a semantic parser for future work. Moreover, other nonsentential sources could be used instead to incorporate semantic information into object codiscovery, for example, inferring the intent or goals of the actors, following their performance of a script, or modeling the physics of the world. Finally, even though our codiscovery results could he used to help train object detectors, we are not suggesting that a natural or fruitful approach to do so would be to annotate training video for object detectors with sentences or predicates. We see our main contribution as proposing a framework for incorporating semantic inference into computer vision.

The remainder of this paper is organized as follows. Section 2 reviews existing codetection methods and other related work, to situate our sentence directed object codiscovery approach. Section 3 presents the details of our approach. Section 4 reports and analyzes extensive experiments. Section 5 discusses the suitability of available datasets for our task. Finally, Sect. 6 concludes with a summary of our contributions.

2 Related Work

Video object codetection is similar to a number of related tasks including image object corecognition, image object codetection, and supervised image/video object detection. Image object corecognition is a simpler variant of image codetection (Tuytelaars et al. 2010), where the objects of interest are sufficiently prominent in the field of view that the problem does not require object localization. Thus corecognition operates like unsupervised clustering, using feature extraction and a similarity measure. Image object codetection (Blaschko et al. 2010; Lee and Grauman 2011; Tang et al. 2014; Cinbis et al. 2014) additionally requires localization, often accomplished by placing bounding boxes around the objects. This can require combinatorial search over a large space of possible object locations (Cinbis et al. 2014). One way to remedy this is to limit the space of possible object locations to those produced by an object proposal method (Alexe et al. 2010; Arbelaez et al. 2014; Zitnick and Dollár 2014; Cheng et al. 2014). These methods typically associate a confidence score with each proposal which can be used to prune or prioritize the search. Codetection is typically formulated as the process of selecting one proposal per image, out of the many produced by the proposal mechanism, that maximizes the collective confidence of and similarity between the selected proposals. This optimization is usually performed with Belief Propagation (Pearl 1982) or with nonlinear programming.

Recently, the codetection problem has been extended to video. Schulter et al. (2013) construct a Conditional Random Field (CRF) in each input video frame with segmented superpixels as vertices. They use both motion and appearance information as unary potentials, and put binary edges between both spatially and temporally neighboring superpixels. Prest et al. (2012) extract motion segments from each video shot to obtain candidate tubes. They then construct a CRF where each video is a vertex and the tubes in the video are the possible labels for that vertex. Joulin et al. (2014) replace the motion tubes with proposal boxes and solve the codetection problem with the Frank–Wolfe algorithm (1956). Srikantha and Gall (2014) incorporate a similarity measure based on an object's functionality in an activity into the codetection framework. Their subsequent work (Srikantha and Gall 2017) extends the approach to select a varying number of tubes for each video and improved the final codetection performance. Ramanathan et al. (2014) develop a bidirectional model for person naming and coreference resolution in video, which is optimized by block coordinate descent. Wang et al. (2014) present a spatio-temporal energy minimization formulation for simultaneous video object discovery and segmentation. Kwak et al. (2015) formulate the codetection process as a combination of establishing correspondences between prominent regions across videos and associating similar object regions within the same video.

Our method closely follows some of the above work. Like Srikantha and Gall (2014, 2017), we codetect small and medium sized objects, but do so without a depth map or heavy dependence on human pose data. Like Schulter et al. (2013), we codetect both moving and stationary objects, but do so with a larger set of object classes and a larger video corpus. Also, like Ramanathan et al. (2014), we use sentences, but do so for a vocabulary that goes beyond pronouns, nominals, and names that are used to codetect only human face tracks.

Most prior work assumes that the objects to be codetected are salient, both in size and appearance, and are located in the center of the field of view (Prest et al. 2012; Joulin et al. 2014; Wang et al. 2014; Kwak et al. 2015). Thus they easily "pop out." For example, limiting codetection to objects in the center of the field of view allowed Prest et al. (2012) to prune the search space by penalizing proposals in contact with the video frame perimeter. Moreover, object proposal methods work well in such simple images, and the confidence score associated with proposals is a reliable measure of salience and a good indicator of which image regions constitute potential objects (Rubinstein et al. 2013). Therefore, the proposal confidence dominates the overall codetection process and the similarity measure only serves to refine the codetection. One exception is Srikantha and Gall (2014, 2017), where they attempt to discover small or medium sized objects in video, without the above simplifying assumptions. However, in order to search through the larger resulting object proposal space, they avail themselves of human pose and depth information to prune the search space.

As a weakly supervised learning problem, image and video object codetection has seen less research effort than fully supervised object detection. While deep learning based object detection methods such as Girshick (2015) are able to detect thousands of object classes, training such requires millions of labeled bounding boxes (Russakovsky et al. 2015). In contrast, our object codiscovery method requires *no* labeled bounding boxes and can codiscover instances of multiple object classes from *tiny* codiscovery sets comprising a few dozen short video clips. As we stated earlier, since codiscovery is a process of ambiguity resolution, it is more difficult and thus more impressive that we can do this with *fewer* videos rather than more videos.

By pairing sentences with video, we formulate video object codiscovery as an *extremely* weakly supervised problem. Sentence semantics only provides ambiguous and implicit labels. This resembles another line of work that learns structured output from image captions (Berg et al. 2004; Gupta and Davis 2008; Luo et al. 2009; Jamieson et al. 2010a, b; Plummer et al. 2015; Mao et al. 2016), treating the input as a parallel image-text dataset. However, all of these methods, except Gupta and Davis (2008) and Jamieson et al. (2010a, b) use *pretrained* object models learned from other datasets. Thus strictly speaking they cannot be called "codetection." Since this line of work focuses on images and not video, the sentential captions only contain static concepts, such as the names of people or the spatial relations between objects in the images. In contrast, our approach models the motion and changing spatial relations that are present only in video as described by verbs and motion prepositions in the sentential annotation. By using sentence semantics to guide codiscovery, we also endow our codiscovery method with the ability for high-level reasoning that is rarely seen in most existing computer-vision methods.

3 Our Approach

Our sentence directed object codiscovery process is illustrated in Fig. 3. The input is a set of videos paired with human-elicited sentences, one sentence per video. For each

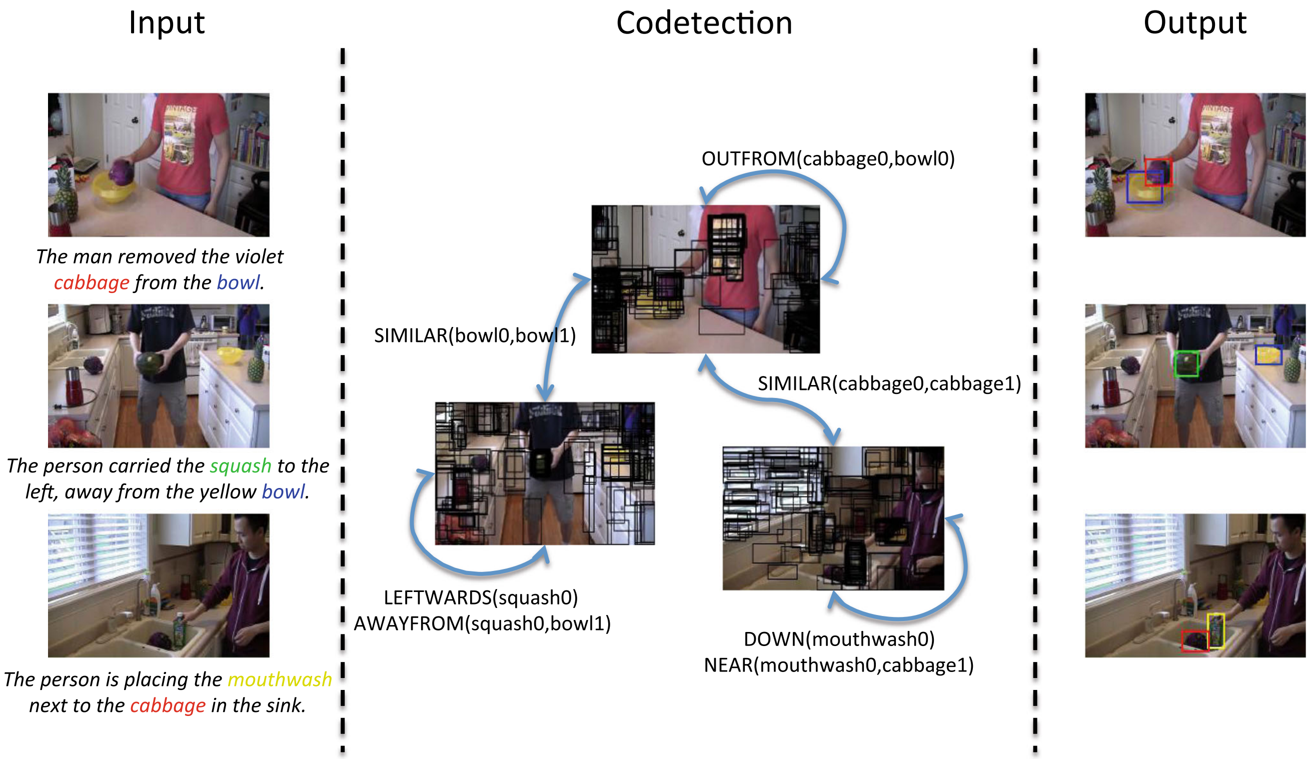

Input
Codetection
Output
The man removed the violet cabbage from the bowl.
The person carried the squash to the left, away from the yellow bowl.
The person is placing the mouthwash next to the cabbage in the sink.
OUTFROM(cabbage0,bowl0)
SIMILAR(bowl0,bowl1)
SIMILAR(cabbage0,cabbage1)
LEFTWARDS(squash0)
AWAYFROM(squash0,bowl1)
DOWN(mouthwash0)
NEAR(mouthwash0,cabbage1)

Fig. 3 An overview of our codiscovery process. *Left* input a set of videos paired with sentences. *Middle* sentence directed object codiscovery, where black bounding boxes represent object proposals. *Right* output original videos with objects codiscovered. Note that no pretrained object detectors are used in this whole process. Also note how sentence semantics plays an important role in this process: it provides both unary scores, e.g., LEFTWARDS(*squash0*) and DOWN(*mouthwash0*), for proposal confidence, and binary scores, e.g., OUTFROM(*cabbage0*, *bowl0*) and NEAR(*mouthwash0*, *cabbage1*), for relating multiple objects in the same video (best viewed in color) (Color figure online)

sentence, we extract a conjunction of predicates together with the object instances as the predicate arguments. In the example from Fig. 3, we have:

The man removed the violet cabbage from the bowl	→	OUTFROM (*cabbage0*, *bowl0*)
The person carried the squash to the left, away from the yellow bowl	→	LEFTWARDS (*squash0* ∧ AWAYFROM (*squash0*, *bowl1*))
The person is placing the mouthwash next to the cabbage in the sink	→	DOWN (*mouthwash0*) ∧ NEAR (*mouthwash0*, *cabbage1*)

The sentences in this example contain six nouns. Thus we extract six object instances: *cabbage0*, *cabbage1*, *squash0*, *bowl0*, *bowl1*, and *mouthwash0*, and produce six tracks, one track per object instance. Two tracks will be produced for each of the three video clips. To accomplish this, a collection of object-candidate generators and video-tracking methods are applied to each video to obtain a pool of object proposals (Sect. 3.2).[1] Any proposal in a video's pool is a possible object instance to assign to a noun in the sentence associated with that video. Given multiple such video-sentence pairs, a graph is formed where object instances serve as vertices and there are two kinds of edges: similarities between object instances and predicates linking object instances in a sentence. Belief Propagation is applied to this graph to jointly infer object codiscoveries by determining an assignment of proposals to each object instance. In the output column of Fig. 3, the red track of the first video clip is selected for *cabbage0*, and the blue track is selected for *bowl0*. The green track of the second video clip is selected for *squash0*, and the blue track is selected for *bowl1*. The red track of the third video clip is selected for *cabbage1*, and the yellow track is selected for *mouthwash0*. All six tracks are produced simultaneously in one inference run. Below, we explain the details of each component of this codiscovery framework.

3.1 Sentence Semantics

Our main contribution is exploiting sentence semantics to help the codiscovery process. We use a conjunction of predicates to represent (a portion of) the semantics of a sentence. Object instances in a sentence fill the arguments of the predicates in that sentence. An object instance that fills the arguments of multiple predicates is said to be *coreferenced*. For a coreferenced object instance, only one track is codiscovered. For example, a sentence like *The person is placing the mouthwash next to the cabbage in the sink* implies the following conjunction of predicates:

DOWN(*mouthwash*) ∧ NEAR(*mouthwash*, *cabbage*)

In this case, *mouthwash* is coreferenced by the predicates DOWN (fills the sole argument) and NEAR (fills the first argument). Thus only one *mouthwash* track will be produced, simultaneously constrained by the two predicates (Fig. 3, yellow track). This coreference mechanism plays a crucial role in the codiscovery process. It tells us that there is exactly one *mouthwash* instance in the above sentence: the *mouthwash* that is being placed down is identical to the one that is placed near the *cabbage*. In the absence of such a coreference constraint, the only constraint between these two potentially different instances of the object class *mouthwash* would be that they are visually similar. Stated informally in English, this would be:

> The cabbage is near a mouthwash that is similar to another mouthwash which is placed down.

Not only does this impose an unnecessarily weaker constraint between *cabbage* and *mouthwash*, it also fails to correctly reflect the sentence semantics.

Following Lin et al. (2014), Kong et al. (2014), and Plummer et al. (2015), our procedure for extracting predicates from a sentence consists of two steps: parsing and transformation/distillation. We first use the Stanford parser (Socher et al. 2013) to parse the sentence. We then employ a set of rules to transform the parsed results to ones that are (1) pertinent to visual analysis, (2) related to a prespecified set of object classes, and (3) distilled so that synonyms are mapped to a common word. These rules simply encode the syntactic variability of how objects fill arguments of predicates. They do not encode semantic information that is particular to specific video clips or datasets. For example, in the sentence *A young man put down the cup*, the adjective *young* is not relevant to our purpose of object codiscovery and will be removed. In the sentence *The person is placing the mouthwash in the sink*, the object *sink* is not one of the prespecified object classes. In this case, we simply ignore the extraneous objects that are out of scope.[2] Thus for the phrase *placing the mouthwash in the sink* in the above sentence, we only extract the predicate DOWN(*mouthwash*). Finally, synonyms introduced by different annotators, e.g., *person*, *man*, *woman*, *child*, and *adult*, are all mapped to a common word (*person*). This mapping process also applies to other parts of speech,

[1] For clarity, in the remainder of this paper, we refer to object proposals for a single frame as object candidates, and object tubes or tracks across a video as object proposals.

[2] We do not attempt to discover objects corresponding to all nouns in any arbitrary sentential description of any arbitrary video. This is obviously beyond the state of the art. Rather, we demonstrate that the constraint provided by sentence semantics can significantly aid video object discovery in many instances.

including verbs, prepositions, and adverbs. This transformation/distillation process never yields *stronger* constraint and usually yields *weaker* constraint than that implied by the semantics of the original sentences.

While we employ a set of manually designed rules, the whole transformation/distillation process is automatic, which allows us to handle sentences of similar structure with the same rule(s). To eliminate the manually designed rules, one could train a semantic parser (Wong and Mooney 2007; Clarke et al. 2010). However, modern semantic parsers are domain specific, and no existing semantic parser has been trained on our domain. Training a new semantic parser usually requires a parallel corpus of sentences paired with intended semantic representations. Semantic parsers are trained with corpora like PropBank (Palmer et al. 2005) that have tens of thousands of manually annotated sentences. Gathering such a large training corpus would be overkill for our experiments that involve only a few hundred sentences, especially since such is not our focus or contribution. Thus we employ simpler handwritten rules to automate the semantic parsing process for our corpora in this paper. Nothing, in principle, precludes using a machine-trained semantic parser in its place. However, we leave that to future work.

The predicates used to represent sentence semantics are formulated around a set of primitive functions on the arguments of the predicate. These produce scores indicating how well the arguments satisfy the constraint intended by the predicate. Figure 4 defines all 36 predicates used to represent sentence semantics in our experiments. Figure 5 defines all 12 primitive functions used to formulate these predicates.

While our predicates are manually designed, they are straightforward to design and code. The effort to do so (several hundred lines of code) could be even less than that of designing a machine learning model that handles the three datasets in our experiments. The reason why this is the case is that the predicates encode only *weak* constraints. Each predicate uses at most four primitive functions (most use only two). The primitive functions are simple, e.g., the temporal coherence (tempCoher) of an object proposal, the average flow magnitude (medFlMg) of a proposal, or simple spatial relations like distLessThan/distGreaterThan between proposals. Unlike features used to support activity recognition or video captioning, these primitive functions need not accurately reflect *every* nuance of motion and changing spatial relations between objects in the video that is implied by the sentence semantics. They need only reflect a weak but sufficient level of the sentence semantics to help guide the search for a reasonable assignment of proposals to nouns during codiscovery. Because of this important property, these primitive functions are not as highly engineered as they might appear to be. Our predicates are general in nature and not specific to specific video samples or datasets.

3.2 Generating Object Proposals

To generate object proposals, we first generate N object candidates for each video frame and construct proposals from these candidates. To support codiscovery of multiple stationary and moving objects, some of which might not be salient and some of which might be occluded for part of the video, our method for generating object candidates must be general purpose: it cannot make assumptions about the video (e.g., simple background) or exhibit bias towards a specific category of objects (e.g., moving objects). Thus methods (Xiao and Lee 2016) that depend on object salience or motion analysis would not suit our purpose. We use EdgeBoxes (Zitnick and Dollár 2014) to obtain the $N/2$ top-ranking object candidates and MCG (Arbelaez et al. 2014) to obtain the other half, filtering out candidates larger than $1/20$ of the video-frame size to focus on small and medium-sized objects. This yields NT object candidates for a video with T frames. We then generate K object proposals from these NT candidates. To obtain object proposals with object candidates of consistent appearance and spatial location, one would nominally require that $K \ll N^T$. To circumvent this, we first randomly sample a frame t from the video with probability proportional to the average magnitude of optical flow (Farnebäck 2003) within that frame. Then, we sample an object candidate from the N candidates in frame t. To decide whether the object is moving or not, we sample from $\{\text{MOVING},\text{STATIONARY}\}$ with distribution $\left\{\frac{1}{3}, \frac{2}{3}\right\}$. We sample a MOVING object candidate with probability proportional to the average flow magnitude within the candidate. Similarly, we sample a STATIONARY object candidate with probability inversely proportional to the average flow magnitude within the candidate. The sampled candidate is then propagated (tracked) bidirectionally to the start and the end of the video. We use the CamShift algorithm (Bradski 1998) to track both MOVING and STATIONARY objects, allowing the size of MOVING objects to change during the process, but requiring the size of STATIONARY objects to remain constant. STATIONARY objects are tracked to account for noise or occlusion that manifests as small motion or change in size. We track STATIONARY objects in RGB color space and MOVING objects in HSV color space. Generally, RGB space is preferable to HSV space because HSV space is noisy for objects with low saturation (e.g., white, gray, or dark) where the hue ceases to differentiate. However, HSV space is used for MOVING objects as it is more robust to motion blur. RGB space is used for STATIONARY objects because motion blur does not arise. We do not use optical-flow-based tracking methods since these methods suffer from drift when objects move quickly. We repeat this sampling and propagation process K times to obtain K object proposals $\{p_k\}$ for each video. Examples of the sampled proposals ($K = 240$) are shown as black boxes in the middle column of Fig. 3.

$$\Delta\textsc{distLarge} \triangleq 0.25 \qquad \Delta\textsc{distSmall} \triangleq 0.05 \qquad \Delta\textsc{angle} \triangleq \pi/2$$

$$\begin{aligned}
\textsc{move}(p) &\triangleq \mathsf{medFlMg}(p)\\
\textsc{moveUp}(p) &\triangleq \textsc{move}(p) + \mathsf{distLessThan}\left(\mathsf{y}(p^{(-L)}) - \mathsf{y}(p^{(L)}), -\Delta\textsc{distLarge}\right)\\
\textsc{moveDown}(p) &\triangleq \textsc{move}(p) + \mathsf{distGreaterThan}\left(\mathsf{y}(p^{(-L)}) - \mathsf{y}(p^{(L)}), \Delta\textsc{distLarge}\right)\\
\textsc{moveVertical}(p) &\triangleq \textsc{move}(p) + \mathsf{distGreaterThan}\left(\left|\mathsf{y}(p^{(-L)}) - \mathsf{y}(p^{(L)})\right|, \Delta\textsc{distLarge}\right)\\
\textsc{moveLeftwards}(p) &\triangleq \textsc{move}(p) + \mathsf{distLessThan}\left(\mathsf{x}(p^{(-L)}) - \mathsf{x}(p^{(L)}), -\Delta\textsc{distLarge}\right)\\
\textsc{moveRightwards}(p) &\triangleq \textsc{move}(p) + \mathsf{distGreaterThan}\left(\mathsf{x}(p^{(-L)}) - \mathsf{x}(p^{(L)}), \Delta\textsc{distLarge}\right)\\
\textsc{moveHorizontal}(p) &\triangleq \textsc{move}(p) + \mathsf{distGreaterThan}\left(\left|\mathsf{x}(p^{(-L)}) - \mathsf{x}(p^{(L)})\right|, \Delta\textsc{distLarge}\right)\\
\textsc{rotate}(p) &\triangleq \textsc{move}(p) + \max_t \mathsf{hasRotation}\left(\mathsf{rotAngle}(p^{(t)}), \Delta\textsc{angle}\right)\\
\textsc{towards}(p_1,p_2) &\triangleq \textsc{move}(p_1) + \mathsf{distLessThan}\left(\mathsf{dist}(p_1^{(-L)}, p_2^{(-L)}) - \mathsf{dist}(p_1^{(L)}, p_2^{(L)}), -\Delta\textsc{distLarge}\right)\\
\textsc{awayFrom}(p_1,p_2) &\triangleq \textsc{move}(p_1) + \mathsf{distGreaterThan}\left(\mathsf{dist}(p_1^{(-L)}, p_2^{(-L)}) - \mathsf{dist}(p_1^{(L)}, p_2^{(L)}), \Delta\textsc{distLarge}\right)\\
\textsc{leftOfStart}(p_1,p_2) &\triangleq \mathsf{tempCoher}(p_2) + \mathsf{distLessThan}\left(\mathsf{x}(p_1^{(L)}) - \mathsf{x}(p_2^{(L)}), -2\Delta\textsc{distSmall}\right)\\
\textsc{leftOfEnd}(p_1,p_2) &\triangleq \mathsf{tempCoher}(p_2) + \mathsf{distLessThan}\left(\mathsf{x}(p_1^{(-L)}) - \mathsf{x}(p_2^{(-L)}), -2\Delta\textsc{distSmall}\right)\\
\textsc{rightOfStart}(p_1,p_2) &\triangleq \mathsf{tempCoher}(p_2) + \mathsf{distGreaterThan}\left(\mathsf{x}(p_1^{(L)}) - \mathsf{x}(p_2^{(L)}), 2\Delta\textsc{distSmall}\right)\\
\textsc{rightOfEnd}(p_1,p_2) &\triangleq \mathsf{tempCoher}(p_2) + \mathsf{distGreaterThan}\left(\mathsf{x}(p_1^{(-L)}) - \mathsf{x}(p_2^{(-L)}), 2\Delta\textsc{distSmall}\right)\\
\textsc{onTopOfStart}(p_1,p_2) &\triangleq \mathsf{tempCoher}(p_2)\\
&\quad + \mathsf{distGreaterThan}\left(\mathsf{y}(p_1^{(L)}) - \mathsf{y}(p_2^{(L)}), -2\Delta\textsc{distLarge}\right)\\
&\quad + \mathsf{distLessThan}\left(\mathsf{y}(p_1^{(L)}) - \mathsf{y}(p_2^{(L)}), 0\right)\\
&\quad + \mathsf{distLessThan}\left(\left|\mathsf{x}(p_1^{(L)}) - \mathsf{x}(p_2^{(L)})\right|, 2\Delta\textsc{distSmall}\right)\\
\textsc{onTopOfEnd}(p_1,p_2) &\triangleq \mathsf{tempCoher}(p_2)\\
&\quad + \mathsf{distGreaterThan}\left(\mathsf{y}(p_1^{(-L)}) - \mathsf{y}(p_2^{(-L)}), -2\Delta\textsc{distLarge}\right)\\
&\quad + \mathsf{distLessThan}\left(\mathsf{y}(p_1^{(-L)}) - \mathsf{y}(p_2^{(-L)}), 0\right)\\
&\quad + \mathsf{distLessThan}\left(\left|\mathsf{x}(p_1^{(-L)}) - \mathsf{x}(p_2^{(-L)})\right|, 2\Delta\textsc{distSmall}\right)\\
\textsc{nearStart}(p_1,p_2) &\triangleq \mathsf{tempCoher}(p_2) + \mathsf{distLessThan}\left(\mathsf{dist}(p_1^{(L)}, p_2^{(L)}), 2\Delta\textsc{distSmall}\right)\\
\textsc{nearEnd}(p_1,p_2) &\triangleq \mathsf{tempCoher}(p_2) + \mathsf{distLessThan}\left(\mathsf{dist}(p_1^{(-L)}, p_2^{(-L)}), 2\Delta\textsc{distSmall}\right)\\
\textsc{inStart}(p_1,p_2) &\triangleq \mathsf{tempCoher}(p_2) + \textsc{nearStart}(p_1,p_2) + \mathsf{smaller}(p_1^{(L)}, p_2^{(L)})\\
\textsc{inEnd}(p_1,p_2) &\triangleq \mathsf{tempCoher}(p_2) + \textsc{nearEnd}(p_1,p_2) + \mathsf{smaller}(p_1^{(-L)}, p_2^{(-L)})\\
\textsc{belowStart}(p_1,p_2) &\triangleq \mathsf{tempCoher}(p_2) + \mathsf{distGreaterThan}\left(\mathsf{y}(p_1^{(L)}) - \mathsf{y}(p_2^{(L)}), \Delta\textsc{distSmall}\right)\\
\textsc{belowEnd}(p_1,p_2) &\triangleq \mathsf{tempCoher}(p_2) + \mathsf{distGreaterThan}\left(\mathsf{y}(p_1^{(-L)}) - \mathsf{y}(p_2^{(-L)}), \Delta\textsc{distSmall}\right)\\
\textsc{aboveStart}(p_1,p_2) &\triangleq \mathsf{tempCoher}(p_2) + \mathsf{distLessThan}\left(\mathsf{y}(p_1^{(L)}) - \mathsf{y}(p_2^{(L)}), -\Delta\textsc{distSmall}\right)\\
\textsc{aboveEnd}(p_1,p_2) &\triangleq \mathsf{tempCoher}(p_2) + \mathsf{distLessThan}\left(\mathsf{y}(p_1^{(-L)}) - \mathsf{y}(p_2^{(-L)}), -\Delta\textsc{distSmall}\right)\\
\textsc{over}(p_1,p_2) &\triangleq \mathsf{tempCoher}(p_2)\\
&\quad + \max_t \left(\begin{array}{l} \mathsf{distLessThan}\left(\mathsf{y}(p_1^{(t)}) - \mathsf{y}(p_2^{(t)}), -\Delta\textsc{distSmall}\right)\\ + \mathsf{distLessThan}\left(\left|\mathsf{x}(p_1^{(t)}) - \mathsf{x}(p_2^{(t)})\right|, \Delta\textsc{distLarge}\right)\end{array}\right)
\end{aligned}$$

$$\begin{aligned}
\textsc{aboveMove}(p) &\triangleq \mathsf{tempCoher}(p) + \textsc{move}(\mathsf{u}(p))\\
\textsc{aroundMove}(p) &\triangleq \mathsf{tempCoher}(p) + \textsc{move}(\mathsf{r}(p))\\
\textsc{inMove}(p) &\triangleq \mathsf{tempCoher}(p) + \textsc{move}(p)\\
\textsc{touchHand}(p) &\triangleq \max_t \mathsf{distLessThan}\left(\mathsf{dist}(p^{(t)} - h^{(t)}), \Delta\textsc{distSmall}\right)\\
\textsc{nearHand}(p) &\triangleq \min_t \mathsf{distLessThan}\left(\mathsf{dist}(p^{(t)} - h^{(t)}), \Delta\textsc{distSmall}\right)\\
\textsc{belowHand}(p) &\triangleq \min_t \mathsf{distGreaterThan}\left(\mathsf{y}(p^{(t)}) - \mathsf{y}(h^{(t)}), 0\right)\\
\textsc{approachHand}(p) &\triangleq \textsc{touchHand}(p) + \mathsf{distLessThan}\left(\mathsf{dist}(p^{(-L)} - h^{(-L)}) - \mathsf{dist}(p^{(L)} - h^{(L)}), -\Delta\textsc{distLarge}\right)
\end{aligned}$$

$$\begin{aligned}
\textsc{behind}(p_1,p_2) &\triangleq \mathsf{tempCoher}(p_2) + \mathsf{distLessThan}\left(\mathsf{y}(p_1) - \mathsf{y}(p_2), -2\Delta\textsc{distSmall}\right)\\
\textsc{frontOf}(p_1,p_2) &\triangleq \mathsf{tempCoher}(p_2) + \mathsf{distGreaterThan}\left(\mathsf{y}(p_1) - \mathsf{y}(p_2), 2\Delta\textsc{distSmall}\right)\\
\textsc{leftOf}(p_1,p_2) &\triangleq \mathsf{tempCoher}(p_2) + \mathsf{distLessThan}\left(\mathsf{x}(p_1) - \mathsf{x}(p_2), -2\Delta\textsc{distSmall}\right)\\
\textsc{rightOf}(p_1,p_2) &\triangleq \mathsf{tempCoher}(p_2) + \mathsf{distGreaterThan}\left(\mathsf{x}(p_1) - \mathsf{x}(p_2), 2\Delta\textsc{distSmall}\right)
\end{aligned}$$

Fig. 4 Our predicates and their semantics. The primitive functions used to compute the predicates are defined in Fig. 5. The symbol p denotes an object proposal, $p^{(t)}$ denotes frame t of an object proposal, and $p^{(L)}$ and $p^{(-L)}$ denote averaging the score of a primitive function over the first and last L frames of a proposal respectively. When there is no time superscript on p, the score is averaged over all frames (e.g., BEHIND). The last eleven predicates only apply to the subset of MPII-Cooking (Sect. 4.2), where h represents the trajectory of the actor's hand over the video frames. Of these, the first seven were added to perform the experiment in Sect. 4.2 and the next four were added to perform the experiment in Sect. 4.3. The portion of the expressions highlighted in *red* indicates that portion used in the FLOW experiment described in Sect. 4.1 (Color figure online)

3.3 Similarity Between Object Proposals

We compute the appearance similarity of two object proposals as follows. We first uniformly sample M boxes $\{b^m\}$ from each proposal p along its temporal extent. For each sampled box, we extract PHOW (Bosch et al. 2007) and HOG (Dalal and Triggs 2005) features to represent its appearance and shape. We also do so after we rotate this detection by 90°, 180°, and 270°. Then, we measure the similarity g between

medFlMg(p) computes the median of the average optical flow magnitude within the detections for proposal p.
x($p^{(t)}$) returns the x-coordinate of the center of $p^{(t)}$, normalized by the frame width and height, respectively.
y($p^{(t)}$) returns the y-coordinate of the center of $p^{(t)}$, normalized by the frame width and height, respectively.
u($p^{(t)}$) returns a neighboring image region above $p^{(t)}$. This image region has the same size as $p^{(t)}$.
r($p^{(t)}$) returns a ring-shape image region around $p^{(t)}$. This image region is four times the size of $p^{(t)}$.
distLessThan(x,a) is defined as $-\log\left(1+\exp(-b(x-a))\right)$. In the experiment $b=-20$.
distGreaterThan(x,a) is defined as $-\log\left(1+\exp(-b(a-x))\right)$.
dist($p_1^{(t)},p_2^{(t)}$) computes the distance between the centers of $p_1^{(t)}$ and $p_2^{(t)}$, normalized by the frame size.
smaller($p_1^{(t)},p_2^{(t)}$) returns 0 if the size of $p_1^{(t)}$ is smaller than that of $p_2^{(t)}$, and $-\infty$ otherwise.
tempCoher(p) evaluates whether the position of proposal p changes during the video by checking the position offsets between every pair of adjacent frames. A higher tempCoher score indicates that p is more likely to be stationary in the video.
rotAngle($p^{(t)}$) computes the current rotation angle of the object inside $p^{(t)}$ by comparing its current orientation with its orientation 1 second (30 frames) earlier in the video. This is computed by extracting SIFT features (Lowe 2004) for both $p^{(t)}$ and $p^{(t-30)}$ and matching them to estimate the similarity transformation matrix, from which the rotation angle can be computed.
hasRotation(α,β) computes the log likelihood of the rotation angle α with the von Mises distribution

$$f(x|\mu,\kappa)=\frac{e^{\kappa\cos(x-\mu)}}{2\pi I_0(\kappa)}$$

taking $\mu=\beta$. In the experiment, the concentration $\kappa=4$.

Fig. 5 The primitive functions used to compute the predicates defined in Fig. 4

a pair of detections b_1^m and b_2^m with:

$$g(b_1^m,b_2^m)=\max_{i,j\in\{0,1,2,3\}}\frac{1}{2}\left(\begin{array}{l}g_{\chi^2}(\mathrm{rot}_i(b_1^m),\mathrm{rot}_j(b_2^m))\\+g_{L_2}(\mathrm{rot}_i(b_1^m),\mathrm{rot}_j(b_2^m))\end{array}\right)$$

where rot_i $i=0,1,2,3$ represents rotation by 0°, 90°, 180°, and 270°, respectively. We use g_{χ^2} to compute the χ^2 distance between the PHOW features and g_{L_2} to compute the Euclidean distance between the HOG features, after which the distances are linearly scaled to [0, 1] and converted to log similarity scores. Finally, the similarity between two proposals p_1 and p_2 is taken to be:

$$g(p_1,p_2)=\operatorname*{median}_m g(b_1^m,b_2^m)$$

3.4 Joint Inference

We extract object instances from the sentences and model them as vertices in a graph. (See all 15 classes for our new dataset, all 5 classes for our subset of CAD-120, and all 7 classes for the subset of MPII-Cooking in Sect. 4.) Each vertex v can be assigned one of the K proposals in the video that is paired with the sentence in which the vertex occurs. The score of assigning a proposal k_v to a vertex v is taken to be the unary predicate score $h_v(k_v)$ computed from the sentence (if such exists, or otherwise 0). We construct an edge between every two vertices u and v that belong to the same object class. We denote this class membership relation as $(u,v)\in\mathscr{C}$. The score of this edge (u,v), when the proposal k_u is assigned to vertex u and the proposal k_v is assigned to vertex v, is taken to be the similarity score $g_{u,v}(k_u,k_v)$ between the two proposals, as described in Sect. 3.3. Similarly, we also construct an edge between two vertices u and v that are arguments of the same binary predicate. We denote this predicate membership relation as $(u,v)\in\mathscr{P}$. The score of this edge (u,v), when the proposal k_u is assigned to vertex u and the proposal k_v is assigned to vertex v, is taken to be the binary predicate score $h_{u,v}(k_u,k_v)$ between the two proposals, as described in Sect. 3.1. Our problem, then, is to select a proposal for each vertex that maximizes the joint score on this graph, i.e., solving the following optimization problem for a CRF:

$$\max_{\mathbf{k}}\sum_v h_v(k_v)+\sum_{(u,v)\in\mathscr{C}}g_{u,v}(k_u,k_v)+\sum_{(u,v)\in\mathscr{P}}h_{u,v}(k_u,k_v)\tag{1}$$

where $\mathbf{k}$ is the collection of the selected proposals for all the vertices. This discrete inference problem can be solved approximately by Belief Propagation (Pearl 1982). In the experiment, we use the OpenGM (Andres et al. 2012) implementation to find the approximate solution.

Conceptually, this joint inference does not require sentences for every video clip. In such a case where some video clips are not described with sentences, we would only have the similarity score g in Eq. 1 for these clips, and would have both the similarity and predicate scores for the rest. This flexibility allows our method to work with videos that do not exhibit apparent semantics or exhibit semantics that can only be captured by extremely complicated predicates or models. Furthermore, our semantic factors h can cooperate with other forms of constraint or knowledge, such as the pose information used by Srikantha and Gall (2014, 2017), by having additional factors in the CRF to encode such constraint or knowledge. Potentially this would further boost the performance of object codiscovery.

4 Experiments

We evaluated our method on three datasets. The first was a newly collected dataset that was filmed in 6 different scenes (four in the KITCHEN, one in the BASEMENT, and one outside the GARAGE) of a house. The lighting conditions vary greatly across the different scenes, with the BASEMENT the darkest, the KITCHEN exhibiting modest lighting, and the GARAGE the brightest. Within each scene, the lighting often varies across different video regions. We assigned 5 actors (four adults and one child) with 15 distinct everyday objects (*bowl*, *box*, *bucket*, *cabbage*, *coffee grinder*, *cooler*, *cup*, *gas can*, *juice*, *ketchup*, *milk*, *mouthwash*, *pineapple*, *squash*, and *watering pot*), and had them perform different actions which involve interaction with these objects. No special instructions were given to the actors; we did not ask them to move slowly or to prevent the objects from being occluded. The actors often are partially outside the field of view.[3] The filming was performed using a normal consumer camera that introduces motion blur on the objects when the actors move quickly. We downsampled the filmed videos to 768 × 432 and divided them into 150 short video clips, each clip depicting a specific event lasting between 2 and 6 s at 30 fps. The 150 video clips constitute a total of 12,509 frames.

The second dataset was a subset of of CAD-120 (Koppula et al. 2013). We constructed this subset by selecting video clips both where individual clips contain multiple object classes and where individual clips contain multiple instances of the same class, to demonstrate the ability of our method to handle both of these situations. The dataset contains 75 clips. These clips have spatial resolution 640 × 480, each clip depicting a specific event lasting between 3 and 5 s at 30 fps. The 75 video clips constitute a total of 8,854 frames, and contain 5 distinct object classes, namely *bowl*, *cereal*, *cup*, *jug*, and *microwave*.

The third dataset was a subset of MPII-Cooking (Rohrbach et al. 2012). The subset contains 233 video clips with spatial resolution 1624 × 1224. Each clip depicts a cooking activity lasting between 1 and 71 s at 30 fps. The 233 video clips constitute a total of 70,259 frames, and contain 7 distinct object classes, namely *bowl*, *bread*, *grater*, *plate*, *spice-holder*, *squeezer*, and *tin*. The original dataset is not provided with sentential annotation. Nonetheless, its video content does describe motion and changing spatial relations between objects. Thus we were able to annotate and use it for comparison with prior work.

These three datasets served to perform two distinct kinds of evaluation with two different objectives: evaluation of novel functionality and comparison with prior work. The first two datasets supported the former while the third dataset supported the latter. We employed the same experimental setup for the first two datasets. This setup evaluated using language to support simultaneous codiscovery of multiple object classes, particularly when individual video clips can depict multiple classes or multiple instances of the same class. We know of no prior work that can do this. Since it was not possible to evaluate this capability via comparison with prior work, we instead compared our full method with four variants that alternatively disable different portions of the scoring function. This helped understand the relative importance of different components of the framework. The third dataset allowed us to perform an apples-to-apples comparison of our method with Prest et al. (2012) and Srikantha and Gall (2014, 2017), particularly measuring the improvement in performance that resulted from the addition of sentential semantics. For this apples-to-apples comparison, like Prest et al. (2012) and Srikantha and Gall (2014, 2017), we only codiscovered one object class at a time, using their predefined codetection splits. In other words, this evaluation did not make use of interaction between object classes during the codiscovery process.

4.1 Our New Dataset and the Subset of CAD-120

Neither our new dataset nor CAD-120 include sentential annotation. Therefore we employed Amazon Mechanical Turk (AMT)[4] to obtain three distinct sentences, by three different workers, for each video clip in each dataset. This yielded 450 sentences for our new dataset and 225 sentences for our subset of CAD-120. AMT annotators were simply instructed to provide a single sentence for each video clip that described the primary activity taking place among objects from a common list of object classes that occur in the entire dataset. The collected sentences were then all converted to the predicates in Fig. 4 using the methods of Sect. 3.1. We processed each of the two datasets three times, each time using a different set of sentences produced by different workers; each sentence was used in exactly one run of the experiment.

As discussed in Sect. 5, our method requires that the dataset contain sufficient linguistic and visual evidence to support codiscovery. If we were to allow annotators to describe *any* object appearing in *any* video clip, the annotation might contain sentences referring to rare objects with insufficient evidence, violating property IV. Or it might contain sentences referring to objects that are too small to be detected by current object proposal mechanisms, violating property III. Thus we provided the list of object classes to

[3] Note that the datasets used by Srikantha and Gall (2014, 2017) do not exhibit this property. Indeed, their method employs human pose which requires that the human be sufficiently visible to estimate such.

[4] https://www.mturk.com/mturk/.

the annotators to focus the annotation on objects that our method can process.

We divided each corpus into codiscovery sets, each set containing a small subset of the video clips. (For our new dataset, some clips were reused in different codiscovery sets. For CAD-120, each clip was used in exactly one codiscovery set.) Three primary criteria were used to split a dataset into codiscovery sets. First, we sought codiscovery sets containing object classes exhibiting mutual spatio-temporal interaction. Without the potential for interaction, joint codiscovery would degenerate into independent codiscovery. For example, *mouthwash* and *gas can* rarely interact so including both in the same codiscovery set would offer no potential to demonstrate joint codiscovery. Second, we sought to ensure that some codiscovery sets contained a mix of videos filmed in different backgrounds. This was done to demonstrate that our method does not rely on simple background modeling (e.g., background subtraction). Finally, we sought to ensure that each codiscovery set contained sufficient visual and linguistic evidence for each object class mentioned as a noun in the sentential annotation for video clips in that codiscovery set.

We processed each codiscovery set for each corpus and each set of sentential annotations (run) independently, each with a distinct CRF. Table 1 summarizes the number of video clips in each codiscovery set for each corpus, the backgrounds contained in each codiscovery set for our new dataset (where K, B, and G denote KITCHEN, BASEMENT, and GARAGE, respectively), and the number of vertices in the resulting CRF for each run of each codiscovery set of each corpus.

We evaluated the resulting codiscovery by measuring the fraction of overlap with human annotation using the standard intersection-over-union (IoU) measure: the ratio of the area of the intersection of two boxes to the area of their union. Human-annotated boxes around objects are provided with CAD-120. For our new dataset, these were obtained with AMT. We obtained five bounding-box annotations for each target object in each video frame. We asked annotators to annotate the referent of a specific highlighted word in the sentence associated with the video containing that frame. Thus the annotation reflects the semantic constraint implied by the sentences. This resulted in $5 \times 289 = 1445$ human annotated tracks. This human annotation exhibits inherent ambiguity: different annotators tend to (1) annotate different parts of an object, and (2) annotate partially occluded objects differently. Informal visual observation of a rendering of the annotated boxes indicated that the second case happens more frequently. To quantify such ambiguity, we computed intercoder agreement between the human annotators for our dataset. We computed $\frac{5\times 4}{2} = 10$ IoU scores for all box pairs produced by the 5 annotators in every frame and averaged them over the entire dataset, obtaining an overall human-human IoU of 0.72.[5]

We compared human annotation against our full method and four variants that alternatively disable different portions of the scoring function in our codiscovery framework, as summarized below:

	SIM	FLOW	SENT	SIM+FLOW	SIM+SENT
					(our full method)
Flow score?	No	Yes	Yes	Yes	Yes
Similarity score?	Yes	No	No	Yes	Yes
Sentence score?	No	Partial	Yes	Partial	Yes

The SIM variant uses the similarity measure but no sentential information. This method is similar to prior video codetection methods that employ similarity and the candidate confidence score output by object candidate generation methods to perform codetection. When the candidate confidence score is not discriminative, as is the case with our datasets, the prior methods degrade to SIM. The FLOW variant exploits only binary movement information from the sentence indicating which objects are probably moving and which are probably not (i.e. using only the functions medFlMg and tempCoher as highlighted in red in Fig. 4), without similarity or any other sentence semantics (thus "partial" in the table). The SIM+FLOW variant adds the similarity score on top of FLOW. The SENT variant uses all possible sentence semantics but no similarity measure. The SIM+SENT variant is our full method that employs all scores. All the above variants were applied to each run of each codiscovery set of each dataset. Except for the changes indicated in the above table, all other parameters were kept constant across all such runs, thus resulting in an apples-to-apples comparison of the results. In particular, $N = 500$, $K = 240$, $M = 20$, and $L = 15$ (see Sect. 3 for details).

We quantitatively evaluated our full method and each of the variants by computing $\text{IoU}_{\text{frame}}$, $\text{IoU}_{\text{object}}$, IoU_{set}, and $\text{IoU}_{\text{dataset}}$ for each variant for each dataset as follows. Given a codiscovered box for an object in a video frame, and the corresponding set of annotated bounding boxes (five boxes for our new dataset and a single box for CAD-120), we computed IoU scores between the codiscovered box and each of the annotated boxes, and took the averaged IoU score as $\text{IoU}_{\text{frame}}$. Then $\text{IoU}_{\text{object}}$ was computed as the average of $\text{IoU}_{\text{frame}}$ over all the codiscovered boxes for the object track, one for each frame. Then IoU_{set} was computed as the average of $\text{IoU}_{\text{object}}$ over all the object tracks in a codiscovery

[5] All datasets reported in this paper, including videos, codetection sets, sentences or predicates, and bounding-box annotation, are available at our project page http://upplysingaoflun.ecn.purdue.edu/~yu239/cccp/sentence-codiscovery.html.

Table 1 The experimental setup of the 10 codiscovery sets for our new dataset and the 5 codiscovery sets for our subset of CAD-120

Scene	Our new dataset, codiscovery set #										Our subset of CAD-120, codiscovery set #				
	1 K1	2 K2	3 K2, K3	4 K4	5 B	6 B	7 G	8 K1, K2, K3	9 B, G	10 K1, K2, K3, K4	1	2	3	4	5
Objects	*Box* *Cabbage* *Coffee grinder* *Mouthwash* *Pineapple* *Squash*	*Bowl* *Cabbage* *Coffee grinder* *Mouthwash* *Pineapple* *Squash*	*Bowl* *Cabbage* *Pineapple* *Squash*	*Cup* *Juice* *Ketchup* *Milk*	*Box* *Cooler*	*Box* *Cooler*	*Bucket* *Gas can* *Watering pot*	*Bowl* *Cabbage* *Pineapple* *Squash*	*Box* *Bucket* *Cooler* *Gas can* *Watering pot*	*Bowl* *Cabbage* *Cup* *Juice* *Ketchup* *Milk* *Mouthwash* *Pineapple*	*Bowl* *Cereal* *Cup* *Jug* *Microwave*	*Bowl* *Cereal* *Cup* *Jug* *Microwave*	*Bowl* *Cereal* *Cup* *Jug* *Microwave*	*Bowl* *Cereal* *Cup* *Jug* *Microwave*	*Bowl* *Cereal* *Cup* *Jug* *Microwave*
# of Videos	26	27	17	21	19	17	23	17	25	24	15	15	15	15	15
# of Vertices in run 1	33	29	24	41	25	26	32	21	35	39	29	27	24	26	27
# of Vertices in run 2	34	37	32	46	24	22	27	26	32	41	25	27	24	22	27
# of Vertices in run 3	33	38	31	36	24	22	33	27	35	39	25	26	21	23	26

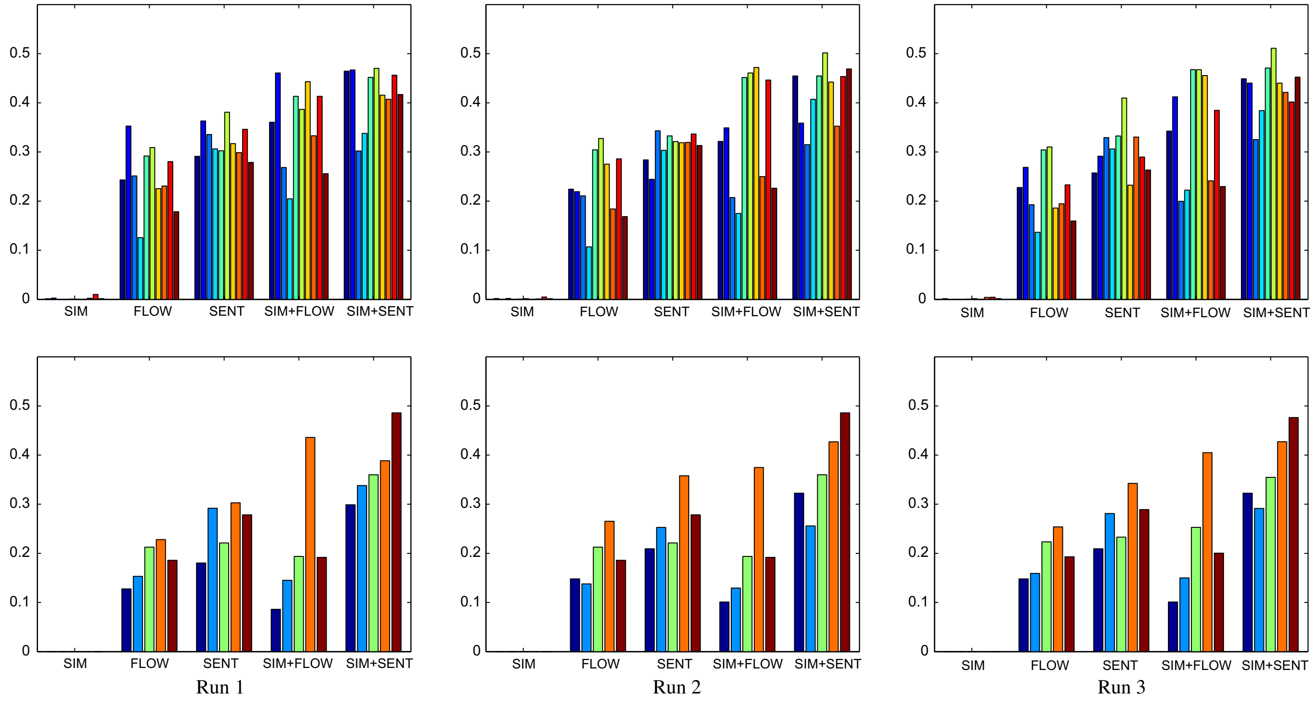


Fig. 6 IoU_{set} scores for all five variants of our method on different runs of different codiscovery sets on two datasets. (top) Our new dataset, codiscovery sets ■ 1, ■ 2, ■ 3, ■ 4, ■ 5, ■ 6, ■ 7, ■ 8, ■ 9, and ■ 10. (*bottom*) Our subset of CAD-120, codiscovery sets ■ 1, ■ 2, ■ 3, ■ 4, and ■ 5

set. Finally $\text{IoU}_{\text{dataset}}$ was computed as the average of IoU_{set} over all runs of all codiscovery sets for a dataset.

For our full method, $\text{IoU}_{\text{dataset}}$ was 0.423 on our new dataset and 0.373 on our subset of CAD-120. We computed IoU_{set} for each variant on each run of each codiscovery set in each dataset as shown in Fig. 6. The first variant, SIM, using only the similarity measure, completely failed on this task as expected. However, combining SIM with either FLOW or SENT improved their performance. Moreover, SENT generally outperformed FLOW, both with and without the addition of SIM. Weak information obtained from the sentential annotation that indicated whether the object was moving or stationary, but no more, i.e. the distinction between FLOW and SENT, was helpful in reducing the object proposal search space, but without the similarity measure, the performance was still quite poor (FLOW). Thus one can get moderate results by combining just SIM and FLOW. But to further boost performance, more sentence semantics is needed, i.e. replacing FLOW with SENT. Further note that for our new dataset, SIM+FLOW outperformed SENT, but for CAD-120, the reverse was true. This seems to be the case because CAD-120 has greater within-class variance so sentential information better supports codiscovery than image similarity. However, over-constrained semantics can, at times, hinder the codiscovery process rather than help, especially given the generality of our datasets. This is exhibited, for example, with codiscovery set 4 (■) on run 1 of the CAD-120 dataset, where SIM+FLOW outperforms SIM+SENT. Thus it is important to only impose *weak* semantics on the codiscovery process.

Also note that there is little variation in IoU_{set} across different runs within a dataset. Recall that the different runs were performed with different sentential annotations produced by different workers on AMT. This indicates that our approach is largely insensitive to the precise sentential annotation.

To evaluate the performance of our method in simply finding objects, we computed codiscovery accuracy $\text{Acc}_{\text{frame}}$, $\text{Acc}_{\text{object}}$, Acc_{set}, and $\text{Acc}_{\text{dataset}}$ for each dataset as follows. Given an IoU threshold, $\text{Acc}_{\text{frame}}$ for a particular codiscovered box in a particular frame was taken to be 1 if $\text{IoU}_{\text{frame}}$ for that box and frame was greater than the threshold, and 0 otherwise. Then $\text{Acc}_{\text{object}}$ was computed as the average of $\text{Acc}_{\text{frame}}$ over all the codiscovered boxes for the object track, one for each frame. Then Acc_{set} was computed as the average of $\text{Acc}_{\text{object}}$ over all the object tracks in a codiscovery set. Finally $\text{Acc}_{\text{dataset}}$ was computed as the average of Acc_{set} over all runs of all codiscovery sets for a dataset. By adjusting the IoU threshold from 0 to 1, we obtained an Acc-vs-threshold curve for each of the variants (Fig. 7). It can be seen that the codiscovery accuracy of our full method is consistently higher than that of the variants under varying IoU thresholds. With IoU thresholds between 0.3 and 0.4, our full method typically yielded codiscovery accuracy (i.e., $\text{Acc}_{\text{dataset}}$) between 0.7 and 0.8

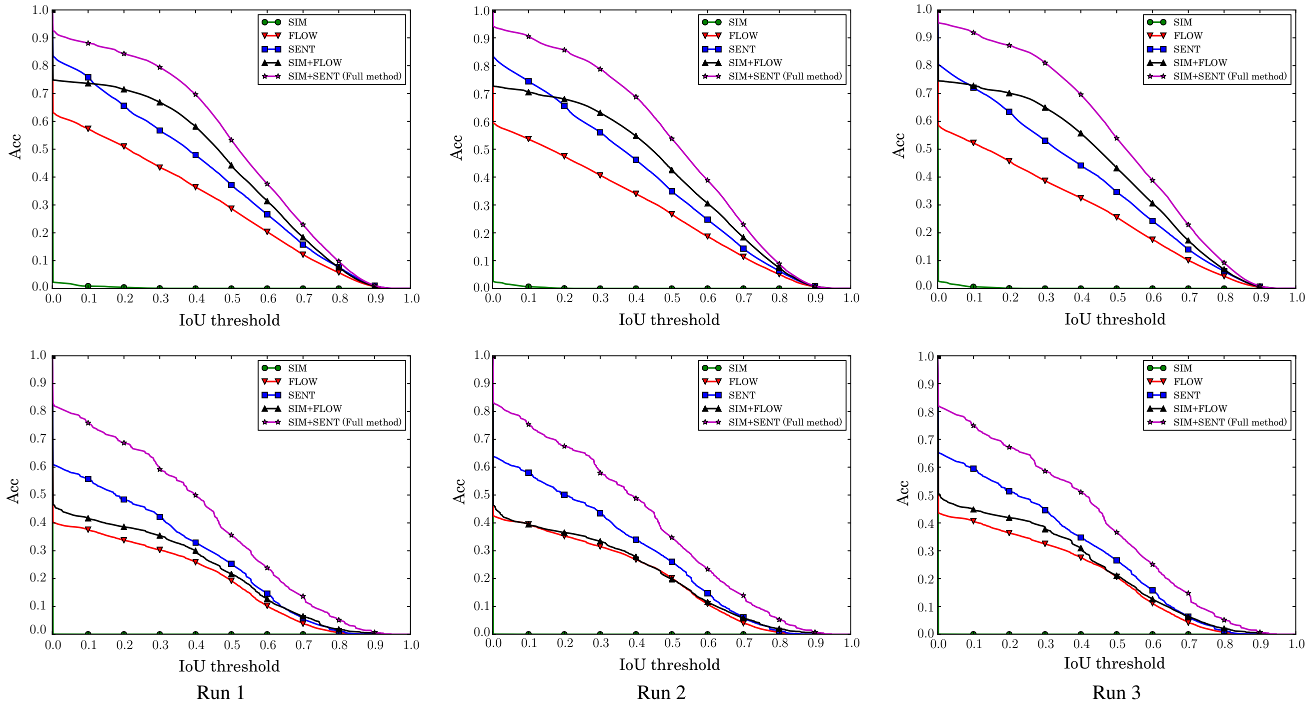


Fig. 7 The codiscovery accuracy curves for all five variants of our method on our new dataset (*top*) and our subset of CAD-120 (*bottom*)

on our new dataset and between 0.5 and 0.6 on our subset of CAD-120. Figure 8 illustrates some codiscovered object examples from both our new dataset and the subset of CAD-120. For more examples, we refer the reader to our project page.[5]

4.2 The Subset of MPII-Cooking

We used the codiscovery sets and human bounding-box annotation from Srikantha and Gall (2014) to process a subset of the MPII-Cooking dataset. The same subset was used by Srikantha and Gall (2014, 2017) for object codetection. The original subset contained 9 object classes. However, Srikantha and Gall (2014, 2017) excluded *pan* and *whisker* from their evaluation, considering only 7 object classes. We did so as well, to perform an apples-to-applies comparison. Furthermore, unlike when using our method to process our new dataset and our subset of CAD-120, where we codiscovered multiple object classes per codiscovery set, each provided codiscovery set for the subset of MPII-Cooking supports codiscovery of a single object class and there is a single codiscovery set for each class. We processed the codiscovery sets independently, each set with a distinct CRF. Table 2 contains the number of video clips in each codiscovery set. Since each codiscovery set supports codiscovery of a single object class, and we only attempted to codiscover a single instance of that class for each clip, the number of vertices in each CRF was equal to the number of videos in that codiscovery set.

The original subset of MPII-Cooking does not include any sentential annotation for the video clips. However, a special property of the subset is that all the video clips in each particular codiscovery set depict the same cooking activity (Table 2). Prest et al. (2012) and Srikantha and Gall (2014, 2017) took advantage of this property to codetect the object that is manipulated by the person in that activity. For example, the *bread* instances to be codetected all participate in the *seasoning bread* activity; the *grater* instances to be codetected all participate in the *grating* activity. We also took advantage of this property to ease the task of pairing video clips with semantic constraint. Since all clips in a codiscovery set depict the same activity and are used to codiscover the same object class, we bypassed the steps of collecting sentential descriptions of the clips from AMT and converting such to predicates and instead annotated all clips in a given codiscovery set with the same manually constructed conjunction of predicates. This allowed as close to an apples-to-apples comparison with Prest et al. (2012) and Srikantha and Gall (2014, 2017) as possible, designed to precisely measure the added benefit of using weak semantic constraint on the codiscovery process.[6] The seven sets of predicates are listed in Table 2.

[6] The MPII-Cooking dataset has a small number of clips relative to the number of distinct object classes that appear in the clips. Eliciting unconstrained sentential descriptions from AMT workers for these clips would run the risk of violating properties I–IV from Sect. 5. Further, sentences that do not describe the target object for each codiscovery set would preclude an apples-to-apples comparison with Prest et al. (2012)

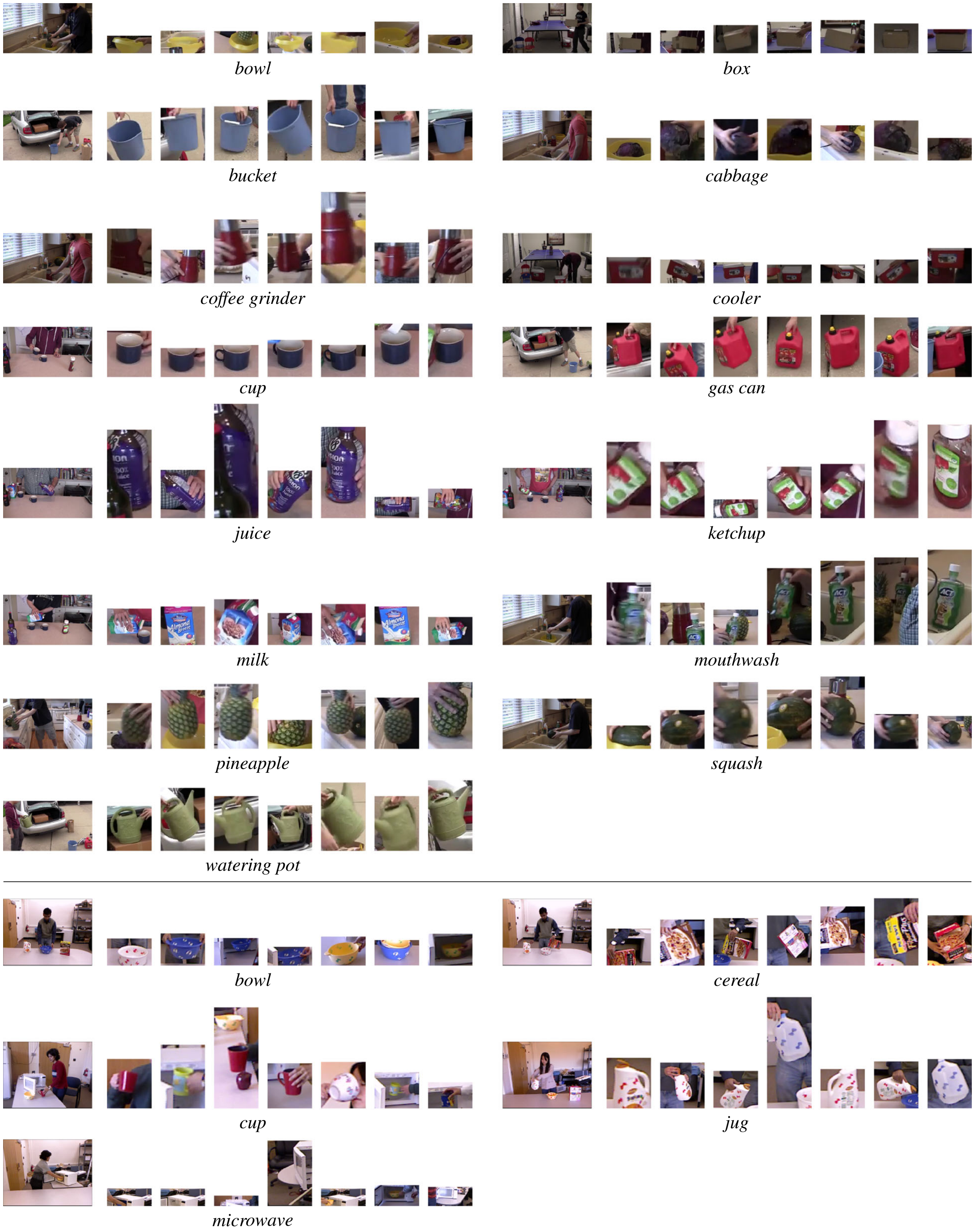


Fig. 8 Examples of the 15 codiscovered object classes in our new dataset (*top*) and the 5 codiscovered object classes in our subset of CAD-120 (*bottom*). The left image for each object class is a sample frame from one of the video clips from which the codiscovery was obtained. While in some examples the objects are occluded, rotated, poorly lit, or blurred due to motion, they are still successfully codiscovered. (For demonstration purposes, the discoveries are slightly enlarged to include the surrounding context. For the quantitative evaluation, the original bounding boxes were used)

Table 2 The experimental setup of the 7 codiscovery sets for the subset of MPII-Cooking

Set #	# of Videos	Object	Activity	Representation of activity in terms of predicates
1	45	*Bowl*	*Filling bowl*	BELOWHAND(p) $\wedge$ INMOVE(p)
2	40	*Bread*	*Seasoning bread*	TOUCHHAND(p) $\wedge$ ABOVEMOVE(p)
3	37	*Grater*	*Grating*	NEARHAND(p) $\wedge$ BELOWHAND(p) $\wedge$ INMOVE(p)
4	41	*Plate*	*Filling plate*	BELOWHAND(p) $\wedge$ INMOVE(p)
5	26	*Spiceholder*	*Fetching spiceholder*	APPROACHHAND(p) $\wedge$ AROUNDMOVE(p)
6	27	*Squeezer*	*Squeezing fruit*	BELOWHAND(p) $\wedge$ ABOVEMOVE(p)
7	17	*Tin*	*Opening tin*	BELOWHAND(p) $\wedge$ NEARHAND(p) $\wedge$ AROUNDMOVE(p)

Only one object class is codiscovered in each codiscovery set. The common cooking activity depicted by the clips in each set is listed. Note that because only one object is codiscovered in each clip, the semantic representation for the activity is constructed from unary predicates that all refer to that same object

Since only a single unambiguous manual semantic annotation was provided for each codiscovery set, only a single run was performed on the subset of MPII-Cooking.

We made two modifications to our codiscovery method to process the subset of MPII-Cooking. First, we employed a different object candidate generation process. Second, we formulated semantic descriptions out of seven new unary predicates (ABOVEMOVE, AROUNDMOVE, INMOVE, TOUCHHAND, NEARHAND, BELOWHAND, and APPROACHHAND), the last four of which constrain the trajectory of the actor's hand (Fig. 4; Table 2). Both of these make use of annotation of the actor's hand position in each frame of each video clip. The MPII-cooking dataset is provided with annotation of the actor's joint positions (including the hand positions) in each frame. Srikantha and Gall (2014, 2017) also avail themselves of this information. Except for these modifications, all other processing was identical to that of our new dataset and the subset of CAD-120.

The first modification was needed because the objects to be codiscovered in this dataset are significantly smaller than those in our new dataset and the subset of CAD-120. The original object candidate generation methods (Zitnick and Dollár 2014; Arbelaez et al. 2014) were unable to reliably produce candidates corresponding to the target codetection objects. Instead, following Srikantha and Gall (2014, 2017), we exploited the estimated human pose information provided with the MPII-Cooking dataset to constrain the search region for object candidates. For each video clip, we limited consideration to a region covering the actor's hand trajectory over the entire clip. Within that region, we considered bounding boxes around each of $N = 200$ superpixels (Achanta et al. 2012) as object candidates. $K = 100$ object proposals were formed from these objects candidates as for the previous two datasets using the same mechanism from Sect. 3.2. The resulting new mechanism was able to reliably produce candidates corresponding to the target codetection object. Note that the modularity of our approach facilitates replacing individual components such as the candidate or proposal generation mechanisms.

The second modification was necessary to provide the semantic primitives needed to encode the kinds of activity depicted in the subset of MPII-Cooking. These were also formulated around the hand trajectory information provided with the dataset and included four new predicates (TOUCHHAND, NEARHAND, BELOWHAND, and APPROACHHAND) that constrain the spatio-temporal relationship between an object proposal and the hand trajectory over the course of a video clip (Fig. 4). Again note that the modularity of our approach facilitates addition of new predicates to enlarge the semantic annotation space.

We compared the resulting codiscoveries against the human annotation provided by Srikantha and Gall (2014). Each video clip is provided with a single human-annotated track of bounding boxes around the target object in each frame. Like before, we evaluated both fraction of overlap with IoU scores and codetection accuracy with Acc, doing so for all five variants of our method (Figs. 9, 10). The same broad pattern of results was obtained as before: (1) SIM completely failed; (2) combining SIM with either FLOW or SENT improved performance; (3) SENT outperformed FLOW both with and without SIM; and (4) SIM+FLOW yielded moderate results, but not as good as SIM+SENT. Figure 11 illustrates some codiscovered object examples from the subset of MPII-Cooking. For more examples, we refer the reader to our project page (see footnote 5).

Table 3 compares our IoU$_{\text{dataset}}$ score with those of Prest et al. (2012) and Srikantha and Gall (2014, 2017), as reported by Srikantha and Gall (2017). It is not possible to provide a breakdown by codetection set (IoU$_{\text{set}}$) as neither Prest et al. (2012) nor Srikantha and Gall (2014, 2017) provide

Footnote 6 continued
and Srikantha and Gall (2014, 2017). Asking the AMT workers to provide sentences that describe interaction with just the target object would likely yield the same semantic description as the ones that we manually constructed as those are the only activities taking place with those object class in the specific codiscovery sets.

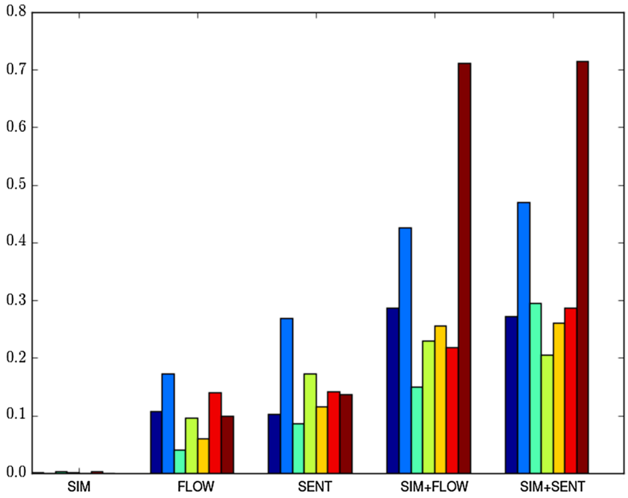


Fig. 9 $\mathrm{IoU}_{\mathrm{set}}$ scores for all five variants of our method on the seven codiscovery sets ■ 1, ■ 2, ■ 3, ■ 4, ■ 5, ■ 6, and ■ 7 of the subset of MPII-Cooking

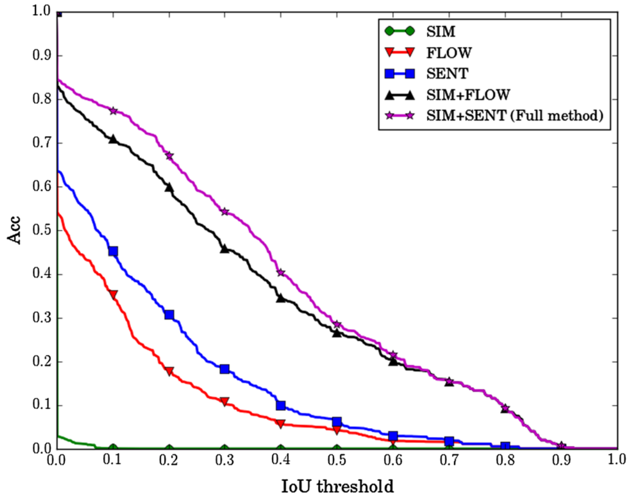


Fig. 10 The codiscovery accuracy curves for all five variants of our method on the subset of MPII-Cooking

such information. Note that our full method, SIM+SENT, by achieving an $\mathrm{IoU}_{\mathrm{dataset}}$ of 0.358, outperformed both Prest et al. (2012) and Srikantha and Gall (2014, 2017), but the variants with only partial scoring functions did not.

4.3 Simultaneous Multi-Class Object Codiscovery on the Subset of MPII-Cooking

The experiment on the subset of MPII-Cooking reported in Sect. 4.2 conducted an apples-to-apples comparison with prior work (Prest et al. 2012; Srikantha and Gall 2014, 2017), but did not showcase the ability of our method to codiscover instances of different classes simultaneously, both within a single video and across a codiscovery set. While we did demonstrate this ability on both our new dataset and the subset of CAD-120, we conducted a further experiment to illustrate this ability on the subset of MPII-Cooking. For this experiment, we added four new predicates (BEHIND, FRONTOF, LEFTOF, and RIGHTOF) to our inventory (Fig. 4) to allow description of the spatial relations between multiple instances of the 7 object classes within individual video clips. We added the new predicates for the 107 out of the 233 clips that depict multiple instances of these 7 object classes within the same clip. Then the whole dataset was partitioned in 10 disjoint codiscovery sets (Table 4) using the following three criteria: (1) each codiscovery set contained at least two different object classes, (2) at least approximately one third of the clips within each set involved activity specified by the new predicates, and (3) the frequencies of the object classes within each set were roughly the same. The only exception to the first criterion above is the tenth set, that only contains *spiceholder*. While there are often several instances of *spice-*

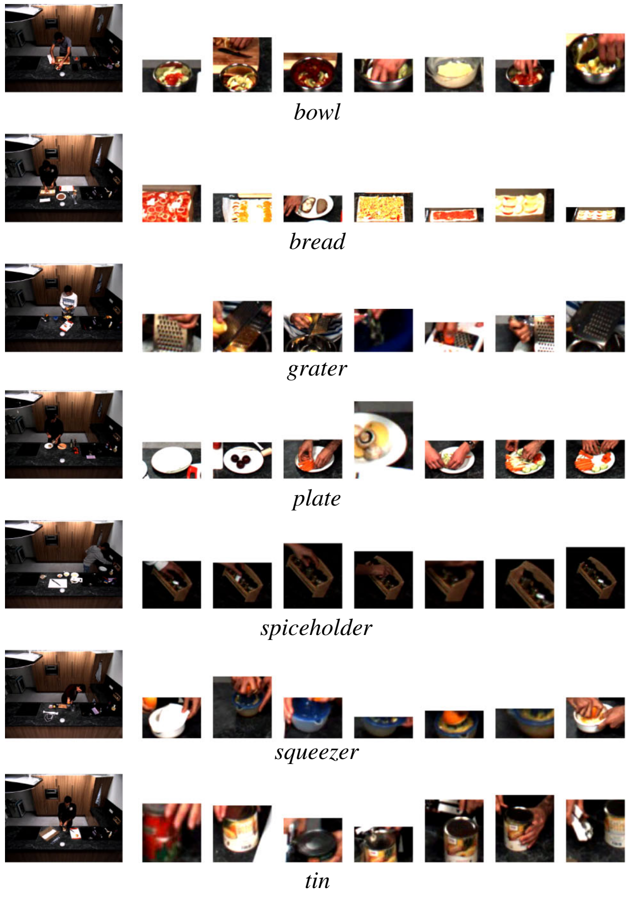


Fig. 11 Examples of the 7 codiscovered object classes in the subset of MPII-Cooking. The *left image* for each object class is a sample frame from one of the video clips from which the codiscovery was obtained. While in some examples the objects are occluded, are blurred, or exhibit high visual variance, they are still successfully codiscovered. (For demonstration purposes, the discoveries are slightly enlarged to include the surrounding context. For the quantitative evaluation, the original bounding boxes were used)

Table 3 $\text{IoU}_{\text{dataset}}$ scores for all five variants of our method along with three prior methods on the subset of MPII-Cooking

	SIM	FLOW	SENT	SIM+FLOW	Prest et al. (2012)	Prest et al. (2012) (Modified)	Srikantha and Gall (2014)	Srikantha and Gall (2017)	SIM+SENT (our full method)
$\text{IoU}_{\text{dataset}}$	0.001	0.102	0.146	0.326	0.023	0.221	0.342	0.348	**0.358**

The modified Prest et al. (2012) employs the tube sampling method of Srikantha and Gall (2014). The original tube generation method of Prest et al. (2012) extracts motion segments as object proposals, which often fails to work on this dataset
Bold value indicates the highest IoU score

Table 4 The experimental setup of the 10 codiscovery sets for the simultaneous multi-class object codiscovery experiment performed on the subset of MPII-Cooking

	Simultaneous multi-class object codiscovery on the subset of MPII-Cooking, codiscovery set #									
	1	2	3	4	5	6	7	8	9	10
Objects	*Bowl* *Bread* *Plate*	*Bowl* *Bread* *Grater* *Plate*	*Bowl* *Plate*	*Plate* *Spiceholder* *Squeezer*	*Bread* *Plate* *Squeezer* *Tin*	*Bowl* *Bread* *Plate*	*Bowl* *Bread* *Plate* *Grater*	*Bowl* *Plate*	*Bowl* *Bread* *Plate* *Tin*	*Spiceholder*
# of Videos	24	25	21	18	29	25	28	22	19	22
# Vertices	36	34	36	29	37	39	37	36	26	30

holder in the same video clip, objects of this class have little interaction with objects of other classes. Since some object classes (e.g., *bowl*, *bread*, and *plate*) have more video clips than others, these classes were assigned to more codiscovery sets. Since the original dataset did not provide human annotation for the bounding boxes of the additional objects in each video, the annotation was augmented with this information. The same method with the same parameters was applied to these codiscovery sets yielding IoU_{set} scores of 0.300, 0.271, 0.309, 0.264, 0.319, 0.412, 0.315, 0.372, 0.361, and 0.215 respectively, and an $\text{IoU}_{\text{dataset}}$ score of 0.314. Figure 12 illustrates some codiscovered object examples from performing simultaneous multi-class object codiscovery on the subset of MPII-Cooking. For more examples, we refer the reader to our project page (see footnote 5). While this problem is far more difficult than that in Sect. 4.2, our method yields multiple objects for nearly half of the video clips, with only slightly lower $\text{IoU}_{\text{dataset}}$ (0.314 *vs.* 0.358). We know of no other existing codetection method that is able to perform this task.

4.4 Failure Cases Analysis

While our object codiscovery method has obtained promising results on all three datasets, it sometimes fails for a variety of reasons (Fig. 13) . One reason is the weak visual feature representation used by our similarity measure: the HOG and PHOW features are limited in their ability to represent and distinguish generic image patches. Thus image regions with similar shapes, colors, or textures could be easily confused. We tried using deep learning features like VggNet (Simonyan and Zisserman 2015). However, the pretrained deep learning models we tried also suffer from this problem. Without fine-tuning or supervised training, these models also can confuse two similar image patches, e.g., the *cutting board* instance in Fig. 13**a** and some of the *bread* instances in Fig. 11. Another failure case occurs when a large object to be codiscovered is always largely occluded in the examples in the codetection set. In this case, the CRF tends to select the unoccluded part of that object in order to satisfy the similarity measure across the different video clips. This occurs, for example, in Fig. 13**b**, where the *microwave* is always occluded by an object like *cup* or *bowl* being placed in or removed from the microwave. The inference algorithm thus selects the upper part of the *microwave* as the final codiscovered result, since it still fits the scoring function. Even though this is not a complete failure, it yields a very low IoU score. Finally, incorrect sentential annotation can negatively impact the codiscovery results. For example, in Fig. 13**c**, the worker annotated a partially incorrect sentence: *The person poured milk next to the ketchup*. While the actor is indeed pouring the *milk*, the *ketchup* is, in fact, far away from the *milk* carton. In this case, the predicate score overpowers the similarity score and a proposal close to the *milk* is selected for *ketchup*. The sentential annotation obtained from AMT for our new dataset and the subset of CAD-120 contains several incorrect sentences.

The above three failure modes appear to account for the majority of the failure cases. Addressing the first failure mode would require a more powerful discriminative feature repre-

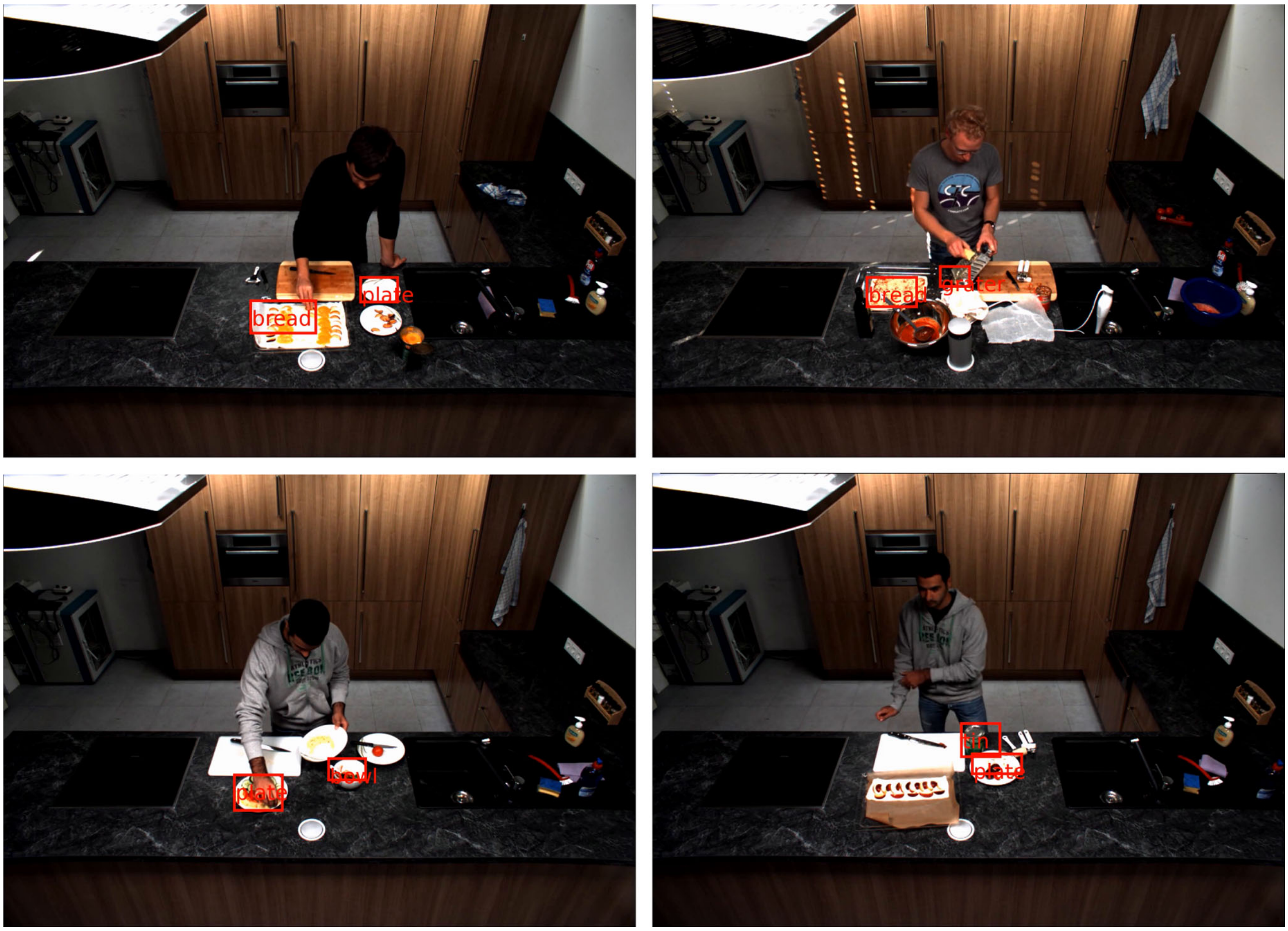
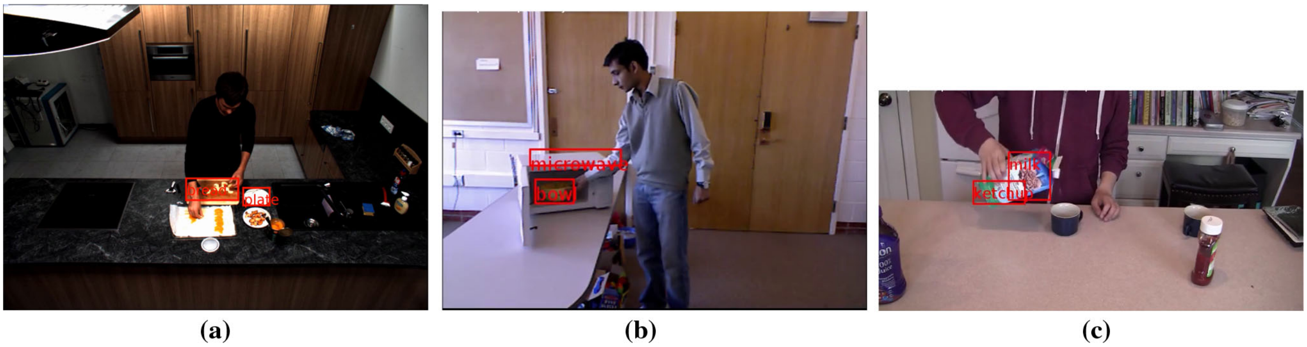


Fig. 12 Examples of codiscovering multiple object classes in a video clip from the subset of MPII-Cooking. When there are several instances of the target object class in the video (e.g., two *plates* in the first example), discovering any of them is considered to be correct



(a) **(b)** **(c)**

Fig. 13 Some failure cases of our codiscovery method. The method fails to work properly on some clips due to: **a** difficulty of discriminating image patches (*bread vs. plate*), **b** large occlusion (*microwave*), and **c** incorrect sentential annotation (*ketchup*). See Sect. 4.4 for details

sentation for image patches, a difficult open problem. The second failure mode might be addressed with a good objectness measure that rewards whole objects and penalizes object parts. But this too is a difficult open problem. The third failure mode might be addressed with techniques to detect and ignore outliers. We leave these nontrivial improvements to future work.

5 Discussion

It is common in computer vision to evaluate new methods by comparison with existing methods on existing datasets. There are a large number of video datasets used for activity recognition and video captioning. There are also datasets used for image and video codetection. The common datasets, however, are ill-suited for the kind of sentence-directed object codiscovery pursued here. Our approach takes advantage of datasets with the following properties:

(I) It applies to video that depicts motion and changing spatial relations between objects.
(II) The video is paired with temporally aligned sentences that describe that motion and those changing spatial relations. Providing such temporal alignment alleviates the need to automatically solve the alignment problem (Zhu et al. 2015; Bojanowski et al. 2015).
(III) The objects to be codiscovered are detectable by existing object proposal methods.
(IV) There is sufficient linguistic and visual evidence across the dataset for each class to be codiscovered. For each object class, the dataset contains a sufficient number of clips that all involve instances of that class participating in the described activity.

It is not possible to evaluate our method on existing image codetection or captioning datasets because they lack property I. It is not possible to evaluate our method on existing video corpora that do not include sentences because they lack property II. While, in principle, it could be possible to augment an existing video corpus with newly collected sentential annotation, as discussed below, not all video corpora are amenable to such.

Sentential annotation is available for some video datasets, like M-VAD (Torabi et al. 2015) and MPII-MD (Rohrbach et al. 2015). However, the vast majority of the clips annotated with sentences in M-VAD (48,986) and MPII-MD (68,337) do not satisfy properties I and II. We searched the sentential annotation provided with each of these two corpora for all instances of twelve common English verbs that represent the kinds of verbs that describe motion and changing spatial relations between objects (Table 5). We further examined ten sentences for each verb from each corpus, together with the corresponding clips, and found that only ten out of the 240 examined satisfied properties I and II. Moreover, none of these ten video clips satisfied property IV.

Table 5 Instances of twelve common English verbs in M-VAD and MPII-MD and the fraction of ten such instances that satisfy properties I and III

	M-VAD		MPII-MD	
Add	89	0/10	120	0/10
Carry	74	1/10	273	2/10
Lift	435	1/10	374	0/10
Load	48	0/10	89	0/10
Move	332	0/10	1106	0/10
Pick	366	1/10	703	1/10
Pour	95	0/10	207	1/10
Put	294	1/10	921	0/10
Rotate	27	0/10	13	0/10
Stack	91	0/10	56	0/10
Take	1058	0/10	1786	0/10
Unload	1	0/10	11	2/10

Sentential annotation is not available for some other video datasets, like Hollywood 2 (Marszałek et al. 2009) and the YouTube-Object dataset used by Prest et al. (2012), Joulin et al. (2014), and Kwak et al. (2015). However, the kinds of activity depicted in the YouTube-Object dataset cannot easily be formulated in terms of descriptions of object motion and changing spatial relations; the video clips usually depict a single object located in the center of the field of view which does not participate in activity that can be described by meaningful sentences that describe its interaction with other objects in the field of view. A similar situation occurs with the Hollywood 2 dataset. Of the twelve classes (*AnswerPhone*, *DriveCar*, *Eat*, *FightPerson*, *GetOutCar*, *HandShake*, *HugPerson*, *Kiss*, *Run*, *SitDown*, *SitUp*, and *StandUp*) in that dataset, only four (*AnswerPhone*, *DriveCar*, *GetOutCar*, and *Eat*) satisfy property I. Of these, three classes (*AnswerPhone*, *DriveCar*, and *GetOutCar*) always depict a single object class, and thus are ill suited for codetecting anything but the two fixed classes *phone* and *car*. The one remaining class (*Eat*) fails to satisfy property IV. This same situation occurs with essentially all standard datasets used for activity recognition, like UCF Sports (Rodriguez et al. 2008) and HMDB51 (Kuehne et al. 2011).

The standard sources of naturally occurring video for corpora used within the computer-vision community are Hollywood movies and YouTube video clips. However, Hollywood movies, in general, mostly involve dialog among actors, or generic scenery and backgrounds. At best, only small portions of most Hollywood movies satisfy property I, and such rarely is reflected in the dialog or script, thus failing to satisfy property II. We attempted to gather a codiscovery corpus from YouTube. But again, about a dozen students searching YouTube for depictions of about a dozen common English verbs, examining hundreds of hits, found that fewer than 1% satisfied property I and none satisfied property IV.

While most existing datasets within the computer-vision community do not satisfy properties I–IV, we believe that these properties are nonetheless reflective of the real natural world. In the real world, people interact with everyday objects (in their kitchen, basement, driveway, and many similar locations) all of the time. It is just that people don't usually record such video, let alone make Hollywood movies about it or post it on YouTube. Further, people rarely describe such in naturally occurring text in movie scripts or in text uploaded to YouTube. Yet, children—and even adults—probably learn names of newly observed objects by observing people in their environment interacting with those objects in the context of dialog about such. Thus we believe that our problem is a natural reflection of the kinds of learning that people employ to learn to recognize newly named objects. Contemporary to our work, Sigurdsson et al. (2016) proposed an interesting new dataset Charades in which hundreds of people record videos in their homes acting casual every activities. We leave the application of our method to this dataset for future work.

It is also instructive to consider our experience extending our method to process the subset of MPII-Cooking. Performing object codiscovery on this dataset is difficult for two reasons. First, the objects are small. Second, there is large within-class variation that weakens the similarity measure. To overcome these challenges, Srikantha and Gall (2014, 2017) exploit human pose data in three ways. First, they take the *functionality* of an object proposal in a cooking activity to be related to its position relative to human pose and assume that object instances of the same class have similar functionality. Second, they employ the distance between the object proposal and the locally active end effector as a unary proposal score, encouraging the selection of the object being manipulated. Third, they explicitly penalize the selection of body parts as codetection solutions, since body parts are dominant in parts of input videos and yield high scores with their scoring functions. These heuristics utilize all eight joints in the human pose information provided with the MPII-Cooking dataset. In contrast, our approach required two simple modifications to process this dataset: focusing a more sensitive candidate generation process on the region around the hands and the addition of four new predicates (TOUCHHAND, NEARHAND, BELOWHAND, and APPROACHHAND) to encode the semantics of the depicted activity in terms of the spatio-temporal relationship between an object proposal and the hand trajectory. Moreover, our approach only utilized the hand trajectory in the human pose information, not the other joints. The modular design of our framework allowed us to easily make these minor modifications. We believe that our approach is more principled, more general, and easier to code than the heuristics employed by Srikantha and Gall (2014, 2017). Moreover, our method significantly outperformed Srikantha and Gall (2014, 2017) despite using less human pose information.

Finally, while our current approach relies on a manually designed semantic parser and manually designed predicates, the general idea of using semantic information about the activity in video to assist object codiscovery is compatible with machine learned semantic parsers and predicates. Automatically learning a parser and predicates for object codiscovery is a challenging yet interesting topic that needs our further investigation in the future. We hope that our current instantiation of the general idea will provide insight for other researchers in this field.

6 Conclusion

We have developed a new framework for object codiscovery in video, namely, using language semantics to guide codiscovery. Our framework is able to simultaneously codiscover multiple object classes and multiple instances of the same class within a single video as well as multiple instances of different classes across video clips on three datasets. Our experiments indicate that weak sentential information can significantly improve the results. This demonstrates that language semantics, when combined with typical computer-vision problems, could provide the capability of high-level reasoning that yields better solutions to these problems.

References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Susstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(11), 2274–2282.

Alexe, B., Deselaers, T., & Ferrari, V. (2010). What is an object? In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 73–80).

Andres, B., Beier, T., & Kappes, J. H. (2012). *OpenGM: A C++ library for discrete graphical models*. CoRR 1206.0111.

Arbelaez, P., Pont-Tuset, J., Barron, J., Marqués, F., & Malik, J. (2014). Multiscale combinatorial grouping. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 328–335).

Barbu, A., Bridge, A., Burchill, Z., Coroian, D., Dickinson, S., & Fidler, S,. et al. (2012). Video in sentences out. In *Proceedings of the conference on uncertainty in artificial intelligence* (pp. 102–112).

Berg, T. L., Berg, A. C., Edwards, J., Maire, M., White, R., Teh, Y. W., Learned-Miller, E. G., & Forsyth, D. A. (2004). Names and faces in the news. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 848–854).

Blaschko, M., Vedaldi, A., & Zisserman, A. (2010). Simultaneous object detection and ranking with weak supervision. In *Advances in neural information processing systems* (pp. 235–243).

Bojanowski, P., Lajugie, R., Grave, E., Bach, F., Laptev, I., Ponce, J., & Schmid, C. (2015). Weakly-supervised alignment of video with text. In *Proceedings of the IEEE international conference on computer vision* (pp. 4462–4470).

Bosch, A., Zisserman, A., & Munoz, X. (2007). Image classification using random forests and ferns. In *Proceedings of the IEEE international conference on computer vision* (pp. 1–8).

Bradski, G.R. (1998). Computer vision face tracking for use in a perceptual user interface, *Intel Technology Journal, Q2*(Q2), 214–219.

Bylinskii, Z., Judd, T., Borji, A., Itti, L., Durand, F., Oliva, A., & Torralba, A. (2012). *MIT saliency benchmark*

Cheng, M. M., Zhang, Z., Lin, W. Y., & Torr, P. (2014). BING: Binarized normed gradients for objectness estimation at 300fps. In

Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3286–3293).

Cinbis, R. G., Verbeek, J., & Schmid, C. (2014). Multi-fold mil training for weakly supervised object localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2409–2416).

Clarke, J., Goldwasser, D., Chang, M. W., & Roth, D. (2010). Driving semantic parsing from the world's response. In *Proceedings of the conference on computational natural language learning* (pp. 18–27).

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 886–893).

Das, P., Xu, C., Doell, R. F., & Corso, J. J. (2013). A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2634–2641).

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*(2), 179–211.

Farnebäck, G. (2003). Two-frame motion estimation based on polynomial expansion. In *Proceedings of the scandinavian conference on image analysis* (pp. 363–370).

Frank, M., & Wolfe, P. (1956). An algorithm for quadratic programming. *Naval Research Logistics*, *3*(1–2), 95–110.

Girshick, R. (2015). Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440–1448).

Graves, A., Wayne, G., & Danihelka, I. (2014). *Neural Turing machines*. CoRR 1410.5401.

Guadarrama, S., Krishnamoorthy, N., Malkarnenkar, G., Venugopalan, S., Mooney, R., Darrell, T., & Saenko, K. (2013). YouTube2Text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *Proceedings of the IEEE international conference on computer vision* (pp. 2712–2719).

Gupta, A., & Davis, L. S. (2008). Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *Proceedings of the european conference on computer vision* (pp. 16–29).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Jamieson, M., Eskin, Y., Fazly, A., Stevenson, S., & Dickinson, S. (2010a) Discovering multipart appearance models from captioned images. In *Proceedings of the European conference on computer vision* (pp. 183–196).

Jamieson, M., Fazly, A., Stevenson, S., Dickinson, S. J., & Wachsmuth, S. (2010b). Using language to learn structured appearance models for image annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(1), 148–164.

Jiang, M., Huang, S., Duan, J., & Zhao, Q. (2015). SALICON: Saliency in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1072–1080).

Joulin, A., Tang, K., & Fei-Fei, L. (2014). Efficient image and video co-localization with Frank-Wolfe algorithm. In *Proceedings of the European conference on computer vision* (pp. 253–268).

Kong, C., Lin, D., Bansal, M., Urtasun, R., & Fidler, S. (2014). What are you talking about? Text-to-image coreference. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3558–3565).

Koppula, H. S., Gupta, R., & Saxena, A. (2013). Learning human activities and object affordances from RGB-D videos. *International Journal of Robotics Research*, *32*(8), 951–970.

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., & Serre, T. (2011). HMDB: A large video database for human motion recognition. In *Proceedings of the IEEE international conference on computer vision* (pp. 2556–2563).

Kwak, S., Cho, M., Laptev, I., Ponce, J., & Schmid, C. (2015). Unsupervised object discovery and tracking in video collections. In *Proceedings of the IEEE international conference on computer vision* (pp. 3173–3181).

Lee, Y. J., & Grauman, K. (2011). Learning the easy things first: Self-paced visual category discovery. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1721–1728).

Lin, D., Fidler, S., Kong, C., & Urtasun, R. (2014). Visual semantic search: Retrieving videos via complex textual queries. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2657–2664).

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Luo, J., Caputo, B., & Ferrari, V. (2009). Who's doing what: Joint modeling of names and verbs for simultaneous face and pose annotation. In *Advances in neural information processing systems* (pp. 1168–1176).

Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A., & Murphy, K. (2016). Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 11–20).

Marszałek, M., Laptev, I., & Schmid, C. (2009). Actions in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2929–2936).

Palmer, M., Gildea, D., & Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, *31*(1), 71–106.

Pearl, J. (1982). Reverend Bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the conference on artificial intelligence* (pp. 133–136).

Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., & Lazebnik, S. (2015). Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision* (pp. 2641–2649).

Prest, A., Leistner, C., Civera, J., Schmid, C., & Ferrari, V. (2012). Learning object class detectors from weakly annotated video. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3282–3289).

Ramanathan, V., Joulin, A., Liang, P., & Fei-Fei, L. (2014). Linking people with "their" names using coreference resolution. In *Proceedings of the european conference on computer vision* (pp. 95–110).

Rodriguez, M. D., Ahmed, J., & Shah, M. (2008). Action MACH: A spatio-temporal maximum average correlation height filter for action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–8).

Rohrbach, A., Rohrbach, M., Qiu, W., Friedrich, A., Pinkal, M., & Schiele, B. (2014). Coherent multi-sentence video description with variable level of detail. In *German conference on pattern recognition* (pp. 184–195).

Rohrbach, A., Rohrbach, M., Tandon, N., & Schiele, B. (2015). A dataset for movie description. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3202–3212).

Rohrbach, M., Amin, S., Andriluka, M., & Schiele, B. (2012). A database for fine grained activity detection of cooking activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1194–1201).

Rubinstein, M., Joulin, A., Kopf, J., & Liu, C. (2013). Unsupervised joint object discovery and segmentation in internet images. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1939–1946).

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Image net large scale visual recognition challenge. *International Journal of Computer Vision*, *115*(3), 211–252.

Schulter, S., Leistner, C., Roth, P. M., & Bischof, H. (2013). Unsupervised object discovery and segmentation in videos. In *Proceedings of the british machine vision conference* (pp. 53.1–53.12).

Sigurdsson, G. A., Varol, G., Wang, X., Farhadi, A., Laptev, I., & Gupta, A. (2016). Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Proceedings of the european conference on computer vision* (pp. 510–526).

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International conference on learning representations*

Socher, R., Bauer, J., Manning, C. D., & Ng, A. Y. (2013). Parsing with compositional vector grammars. In *Proceedings of the annual meeting of the association for computational linguistics* (pp. 455–465).

Srikantha, A., & Gall, J. (2014). Discovering object classes from activities. In *Proceedings of the european conference on computer vision* (pp. 415–430).

Srikantha, A., & Gall, J. (2017). Weak supervision for detecting object classes from activities. *Computer Vision and Image Understanding.*, *156*(C), 138–150.

Tang, K., Joulin, A., Li, J., & Fei-Fei, L. (2014). Co-localization in real-world images. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1464–1471).

Torabi, A., Chris, P., Hugo, L., & Aaron, C. (2015). Using descriptive video services to create a large data source for video annotation research. CoRR 1503.01070

Tuytelaars, T., Lampert, C. H., Blaschko, M. B., & Buntine, W. L. (2010). Unsupervised object discovery: A comparison. *International Journal of Computer Vision*, *88*(2), 284–302.

Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R. J., & Saenko, K. (2015). Translating videos to natural language using deep recurrent neural networks. In *The conference of the North American chapter of the association for computational linguistics: Human language technologies* (pp. 1494–1504).

Wang, L., Hua, G., Sukthankar, R., Xue, J., & Zheng, N. (2014). Video object discovery and co-segmentation with extremely weak supervision. In *Proceedings of the european conference on computer vision* (pp. 640–655).

Wong, Y. W., & Mooney, R. J. (2007). Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of the annual meeting of the association for computational linguistics* (pp. 960–967).

Xiao, F., & Lee, Y. J. (2016). Track and segment: An iterative unsupervised approach for video object proposals. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 933–942).

Yu, H., Siddharth, N., Barbu, A., & Siskind, J. M. (2015). A compositional framework for grounding language inference, generation, and acquisition in video. *Journal of Artificial Intelligence Research*, *52*, 601–713.

Yu, H., Wang, J., Huang, Z., Yang, Y., & Xu, W. (2016). Video paragraph captioning using hierarchical recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4584–4593).

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision* (pp. 19–27).

Zitnick, C. L., & Dollár, P. (2014). Edge boxes: Locating object proposals from edges. In *Proceedings of the european conference on computer vision* (pp. 391–405).