

Grounded Human-Object Interaction Hotspots from Video

Tushar Nagarajan*
UT Austin

tushar@cs.utexas.edu

Christoph Feichtenhofer
Facebook AI Research

feichtenhofer@fb.com

Kristen Grauman
Facebook AI Research

grauman@fb.com[†]

Abstract

Learning how to interact with objects is an important step towards embodied visual intelligence, but existing techniques suffer from heavy supervision or sensing requirements. We propose an approach to learn human-object interaction “hotspots” directly from video. Rather than treat affordances as a manually supervised semantic segmentation task, our approach learns about interactions by watching videos of real human behavior and anticipating afforded actions. Given a novel image or video, our model infers a spatial hotspot map indicating how an object would be manipulated in a potential interaction—even if the object is currently at rest. Through results with both first and third person video, we show the value of grounding affordances in real human-object interactions. Not only are our weakly supervised hotspots competitive with strongly supervised affordance methods, but they can also anticipate object interaction for novel object categories. Project page: <http://vision.cs.utexas.edu/projects/interaction-hotspots/>

1. Introduction

Today’s visual recognition systems know how objects *look*, but not how they *work*. Understanding how objects function is fundamental to moving beyond passive perceptual systems (e.g., those trained for image recognition) to active, embodied agents that are capable of both perceiving and interacting with their environment—whether to clear debris in a search and rescue operation, cook a meal in the kitchen, or even engage in a social event with people. Gibson’s theory of affordances [17] provides a way to reason about object function. It suggests that objects have “action possibilities” (e.g., a chair affords sitting, a broom affords cleaning), and has been studied extensively in computer vision and robotics in the context of action, scene, and object understanding [22].

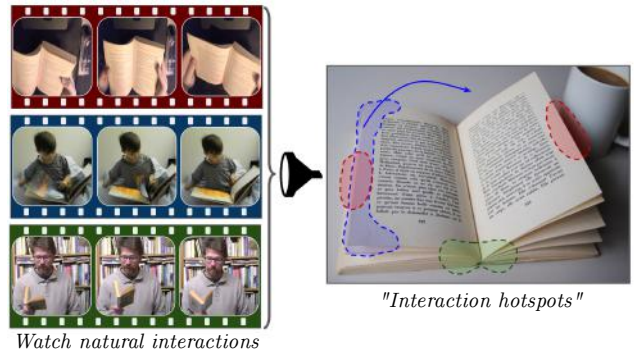


Figure 1: We propose to learn object affordances directly from videos of people naturally interacting with objects. The resulting representation of “interaction hotspots” is grounded in real human behavior from video, rather than manual image annotations.

However, the abstract notion of “what actions are possible?” is only half the story. For example, for an agent tasked with sweeping the floor with a broom, knowing that the broom handle *affords holding* and the broom *affords sweeping* is not enough. The agent also needs to know *how* to interact with different objects, including the best way to grasp the object, the specific points on the object that need to be manipulated for a successful interaction, how the object is used to achieve a goal, and even what it suggests about how to interact with *other* objects.

Learning how to interact with objects is challenging. Traditional methods face two key limitations. First, methods that consider affordances as properties of an object’s shape or appearance [37, 18, 24] fall short of modeling actual object use and manipulation. In particular, learning to segment specified object parts [38, 49, 37, 39] can capture annotators’ expectations of what is important, but is detached from real interactions, which are dynamic, multi-modal, and may only partially overlap with part regions (see Figure 1). Secondly, existing methods are limited by their heavy supervision and/or sensor requirements. They assume access to training images with manually drawn masks or keypoints [46, 10, 12] and some leverage additional sensors like depth [31, 66, 67] or force gloves [3], all of which

*Work done during internship at Facebook AI Research.

[†]On leave from UT Austin (grauman@cs.utexas.edu).

restrict scalability. Such bottlenecks also deter generalization: exemplars are often captured in artificial lab tabletop environments [37, 31, 49] and labeling cost naturally restricts the scope to a narrow set of objects.

In light of these issues, we propose to learn affordances that are *grounded* in real human behavior directly from videos of people naturally interacting with objects, and without any keypoint or mask supervision. Specifically, we introduce an approach to infer an object’s *interaction hotspots*—the spatial regions most relevant to human-object interactions. Interaction hotspots link *inactive* objects at rest not only to the actions they afford, but also to *how* they afford them. By learning hotspots directly from video, we sidestep issues stemming from manual annotations, avoid imposing part labels detached from real interactions, and discover exactly how people interact with objects in the wild.

Our approach works as follows. First, we use videos of people performing everyday activities to learn an action recognition model that can recognize the array of afforded actions when they are *actively in progress* in novel videos. Then, we introduce an anticipation model to distill the information from the video model, such that it can estimate how a static image of an *inactive* object transforms during an interaction. In this way, we learn to anticipate the plausible interactions for an object at rest (*e.g.*, perceiving “cuttable” on the carrot, despite no hand or knife being in view). Finally, we propose an activation mapping technique tailored for fine-grained object interactions to derive interaction hotspots from the anticipation model. Thus, given a new image, we can hypothesize interaction hotspots for an object, even if it is not being actively manipulated.

We validate our model on two diverse video datasets: OPRA [12] and EPIC-Kitchens [7], spanning hundreds of object and action categories, with videos from both first and third person viewpoints. Our results show that with just weak action and object labels for training video clips, our interaction hotspots can predict object affordances more accurately than prior weakly supervised approaches, with relative improvements up to 25%. Furthermore, we show that our hotspot maps can anticipate object function for novel object classes that are never seen during training, and that our model’s learned representation encodes functional similarities between objects that go beyond appearance features.

In summary, we make the following contributions:

- We present a framework that integrates action recognition, a novel anticipation module, and feature localization to learn object affordances directly from video, without manually annotated segmentations/keypoints.
- We propose a class activation mapping strategy tailored for fine-grained object interactions that can learn high resolution, localized activation maps.
- Our approach predicts affordances more accurately than prior weakly supervised methods—and even competitively with strongly supervised methods—and can anticipate object interaction for novel object classes unobserved in the training video.

2. Related Work

Visual Affordances. The theory of affordances [17], originally from work in psychology, has been adopted to study several tasks in computer vision [22]. In action understanding, affordances provide context for action anticipation [32, 45, 65] and help learn stronger action recognition models [30]. In scene understanding, they help decide *where* in a scene a particular action can be performed [47, 18, 60, 9], learn scene geometry [21, 15], or understand social situations [5]. In object understanding, affordances help model object function and interaction [53, 62, 67], and have been studied jointly with hand pose/configuration [29, 54, 3] and object motion [19, 20].

The choice of affordance representation varies significantly in these tasks, spanning across human pose, trajectories of objects, sensorimotor grasps, and 3D scene reconstructions. Often, this results in specialized hardware and heavy sensor requirements (*e.g.*, force gloves, depth cameras). We propose to automatically learn appropriate representations for visual affordances directly from RGB video of human-object interactions.

Grounded Affordances. Pixel-level segmentation of object parts [49, 37, 39] is a common affordance representation, for which supervised semantic segmentation frameworks are the typical approach [37, 46, 39, 10]. These segmentations convey high-level information about object function, but rely on manual mask annotations to train—which are not only costly, but can also give an unrealistic view of how objects are actually used. Unlike our approach, such methods are “ungrounded” in the sense that the annotator declares regions of interest on the objects outside of any interaction context.

Representations that are grounded in human behavior have also been explored. In images, human body pose serves as a proxy for object affordance to reveal modes of interaction with musical instruments [62, 63] or likely object interaction regions [4]. Given a video, methods can parse 3D models to estimate physical concepts (velocity, force, etc.) in order to categorize object interactions [66, 67]. For instructional video, methods explore ways to extract object states [1], modes of object interaction [8], interaction regions [12], or the anticipated trajectory of an object given a person’s skeleton pose [31].

We introduce a new approach for learning affordance “heatmaps” grounded in human-object interaction, as derived directly from watching real-world videos of people

using the objects. Our model differs from other approaches in two main ways. First, no prior about interaction in the form of human pose, hand position, or 3D object reconstruction is used. All information about the interactions is learned directly from video. Second, rather than learn from manually annotated ground truth masks or keypoints [37, 46, 39, 10, 49, 48, 12], our model uses only coarse action labels for video clips to guide learning.

Video anticipation. Predicting future frames in videos has been studied extensively in computer vision [43, 36, 58, 35, 52, 55, 59, 40, 61, 57, 28]. Future prediction has been applied to action anticipation [26, 56, 32, 44], active-object forecasting [16], and to guide demonstration learning in robotics [13, 14, 11]. In contrast to these works, we devise a novel anticipation task—learning object interaction affordances from video. Rather than predict future frames or action labels, our model anticipates correspondences between *inactive* objects (at rest, and not interacted with) and *active* objects (undergoing interaction) in feature space, which we then use to estimate affordances.

3. Approach

Our goal is to learn “interaction hotspots”: characteristic object regions that anticipate and explain human-object interactions (see Figure 1). Conventional approaches for learning affordance segmentation only address part of this goal. Their manually annotated segmentations are expensive to obtain, do not capture the dynamics of object interaction, and are based on the annotators’ notion of importance, which does not always align with real object interactions. Instead of relying on such segmentations as proxies for interaction, we train our model on a more direct source—videos of people naturally interacting with objects. We contend that such videos contain much of the cues necessary to piece together *how* objects are interacted with.

Our approach consists of three steps. First, we train a video action classifier to recognize each of the afforded actions (Section 3.1). Second, we introduce a novel anticipation model that maps static images of the inactive object to its afforded actions (Section 3.2). Third, we propose an activation mapping technique in the joint model tailored for discovering interaction hotspots on objects, without any keypoint or segmentation supervision (Section 3.3). Given a static image of a novel object, we use the learned model to extract its *hotspot hypotheses* (Section 3.4). Critically, the model can infer hotspots even for objects unseen during training, and regardless of whether the object is actively being interacted with in the test image.

3.1. Learning Afforded Actions from Video

Our key insight is to learn about object interactions from video. In particular, our approach learns to predict afforded

actions across a span of objects, then translates the video cues to static images of an object at rest. In this way, without explicit region labels and without direct estimation of physical contact points, we learn to anticipate object use. Throughout, we use the term “active” to refer to the object when it is involved in an interaction (*i.e.*, the status during training) and “inactive” to refer to an object at rest with no interaction (*i.e.*, the status during testing).

Let \mathcal{A} denote the set of all afforded actions (*e.g.*, *pourable*, *pushable*, *cuttable*), and let \mathcal{O} denote the set of object categories (*e.g.*, *pan*, *chair*, *blender*), each of which affords one or more actions in \mathcal{A} . During training, we have video clips containing various combinations of afforded actions and objects.

First, we train a video-classification model to predict which afforded action occurs in a video clip. For a video of T frames $\mathcal{V} = \{f_1, \dots, f_T\}$ and afforded action class a , we encode each frame using a convolutional neural network backbone to yield $\{x_1, \dots, x_T\}$. Each x_t is a tensor with d channels, each with an $n \times n$ spatial extent, with d and n determined by the specific backbone used.¹ These features are then spatially pooled to obtain a d -dimensional vector per frame:

$$g_t(x_t) = P(x_t) \quad \text{for } t = 1, \dots, T, \quad (1)$$

where P denotes the L2-pooling operator. We describe the choice to use this over traditional average pooling in Section 3.3.

We further aggregate the frame-level features over time,

$$h_*(\mathcal{V}) = \mathbb{A}(g_1, \dots, g_T), \quad (2)$$

where \mathbb{A} is a *video aggregation module* that combines the frame features of a video into an aggregate feature h_* for the whole video. In our experiments, we use a long short-term memory (LSTM) recurrent neural network [25] for \mathbb{A} . We note that our framework is general and other video classification architectures (*e.g.*, 3D ConvNets) can be used.

The aggregate video feature h_* is then fed to a linear classifier to predict the afforded action, which is trained using cross-entropy loss $\mathcal{L}_{cls}(h_*, a)$. Once trained, this model can predict which action classes are observed in a video clip of arbitrary length. See Figure 2 (left) for the architecture.

Note that the classifier’s predictions are *object category-agnostic*, since we train it to recognize an afforded action across instances of any object category that affords that action. In other words, the classifier knows $|\mathcal{A}|$ total actions, not $|\mathcal{A}| \times |\mathcal{O}|$; it recognizes *pourable* + X as one entity, as opposed to *pourable* + *cup* and *pourable* + *bowl* separately. This point is especially relevant once we leverage the model below to generalize hotspots to unfamiliar object classes.

¹For example, our experiments use a modified ResNet [23] backbone, resulting in $d = 2048$ and $n = 28$, or a $2048 \times 28 \times 28$ feature per frame.

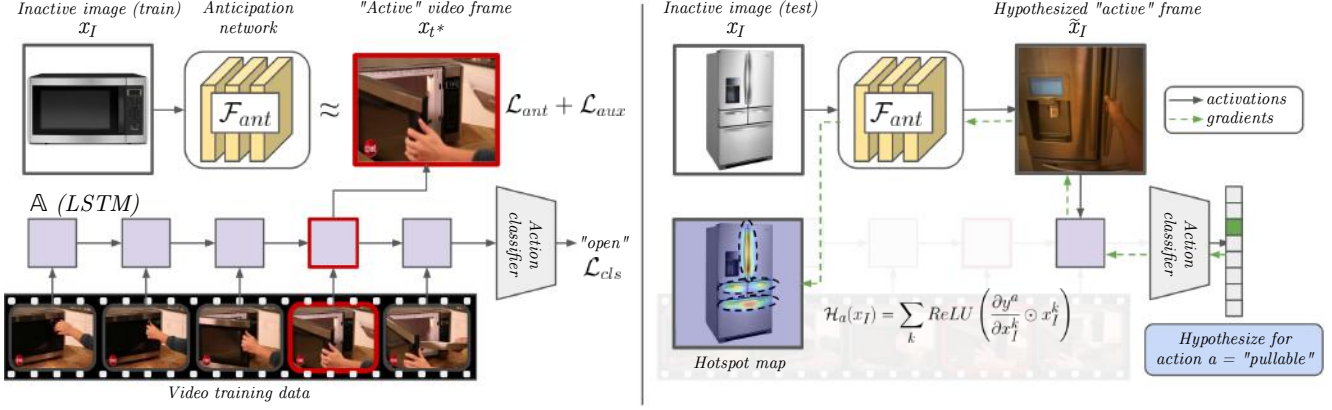


Figure 2: Illustration of our framework for training (left) and testing (right). **Left panel:** The two components of our model—the video action classifier (Sec. 3.1) and the anticipation module with its associated losses (Sec. 3.2 and 3.3)—are jointly trained to predict the action class in a video clip while building an affordance-aware internal representation for objects. **Right panel:** Once trained, our model generates “interaction hotspot” maps for a novel *inactive* object image (top left fridge image). It first hallucinates features that would occur for the object *if it were active* (top right photo), then derives gradient-weighted attention maps over the original image, for each action. Our method can infer hotspots even for novel object categories unseen in the training video; for example, learning about opening microwaves helps anticipate how to open the fridge. Note that x_I, \tilde{x}_I are in feature space, not pixel space.

3.2. Anticipation for Inactive Object Affordances

So far, we have a video recognition pipeline that can identify occurrences of the afforded actions on sequences with active human-object interactions. This model alone would focus on “active” cues directly related to the action being performed (*e.g.*, hands approaching an object), but would not respond strongly to *inactive* instances—static images of objects that are at rest and not being interacted with. In fact, prior work demonstrates that these two incarnations of objects are visually quite different, to the point of requiring distinct object detectors, *e.g.*, to recognize both open and closed microwaves [42].

We instead aim for our system to learn about object affordances by watching video of people handling objects, then mapping that knowledge to novel inactive object photos/frames. To bridge this gap, we introduce a distillation-based anticipation module \mathcal{F}_{ant} that transforms the embedding of an inactive object x_I , where no interaction is occurring, into its active state where it is being interacted with:

$$\tilde{x}_I = \mathcal{F}_{ant}(x_I). \quad (3)$$

See Figure 2, top-left. In experiments we consider two avenues for obtaining the inactive object training images x_I : inactive frames from a training sequence showing the object before an interaction starts, or catalog photos of the object shown at rest. During training, the anticipation module is guided by the video action classifier, which selects the appropriate *active state* from a given video as the frame x_{t^*} at which the LSTM is maximally confident of the true action:

$$t^* = \arg \min_{t \in 1..T} \mathcal{L}_{cls}(\mathbb{A}(g_1, \dots, g_t), a), \quad (4)$$

where a is the true afforded action label, and \mathcal{L}_{cls} is again the cross-entropy loss for classification using the aggregated hidden state at each time t .

We then define a feature matching loss between (a) the anticipated active state for the inactive object and (b) the active state selected by the classifier network for the training sequence. This loss requires the anticipation model to hypothesize a grounded representation of what an object would look like during interaction, according to the actual training video:

$$\mathcal{L}_{ant}(x_I, x_{t^*}) = \|P(\tilde{x}_I) - P(x_{t^*})\|_2. \quad (5)$$

Additionally, we make sure that the newly anticipated representation \tilde{x}_I is predictive of the afforded action and compatible with our video classifier, by using it to classify the afforded action after a single step of the LSTM, resulting in an auxiliary classification loss $\mathcal{L}_{aux}(h_1(\tilde{x}_I), a)$.

Overall, these components allow our model to estimate what a static inactive object may potentially look like—in feature space—if it were to be interacted with. They provide a crucial link between classic action recognition and affordance learning. As we will define next, activation mapping through \mathcal{F}_{ant} then provides information about what spatial locations on the original static image are most strongly correlated to how it would be interacted with.

3.3. Interaction Hotspot Activation Mapping

At test time, given an inactive object image q , our goal is to infer the *interaction hotspot* maps \mathcal{H}_a , for all $a \in \mathcal{A}$, each of which is an $H \times W$ matrix summarizing the regions of interest that characterize an object interaction, where H, W



Figure 3: Our method (bottom) vs. traditional action recognition+Grad-CAM (top). Our model generates localized affordance-relevant activations, which a standard action recognition model is unable to do.

denote the height and width of the source image.² Intuitively, a hotspot map should pick up on the regions of the object that would be manipulated or otherwise transform during the action a , indicative of its affordances. Note that there is one map per action $a \in \mathcal{A}$.

We devise an activation mapping approach to go from inactive image embeddings x_I to interaction predictions $\mathcal{F}_{ant}(x_I)$, and finally to hotspots, tailoring it for discovering our hotspot maps. For a particular inactive image embedding x_I and an action a , we compute the gradient of the score for the action class with respect to each channel of the embedding. These gradients are used to weight individual spatial activations in each channel, acting as an attention mask over them. The positive components of the resulting tensor are retained and accumulated over all channels in the input embedding to give the final hotspot map $\mathcal{H}_a(x_I)$ for the action class:

$$\mathcal{H}_a(x_I) = \sum_k \text{ReLU} \left(\frac{\partial y^a}{\partial x_I^k} \odot x_I^k \right), \quad (6)$$

where x_I^k is the k^{th} channel of the input frame embedding and \odot is the element-wise multiplication operator. This is meaningful only when the gradients are not spatially uniform (e.g., not if x_I is average pooled for classification). We use L2-pooling to ensure that spatial locations produce gradients as a function of their activation magnitudes.

Next, we address the spatial resolution. The reduced spatial resolution from repeatedly downsampling features in the typical ResNet backbone is reasonable for classification, but is a bottleneck for learning interaction hotspots. We set the spatial stride of the last two residual stages to 1 (instead of 2), and use a dilation for its filters. This increases the spatial resolution by $4 \times$ to $n = 28$, allowing our heatmaps to capture finer details.

Our technique is related to other feature visualization methods [51, 64, 50]. However, we use a reduced stride and L2 pooling to make sure that the gradients themselves are spatially localized, and like [51], we do not spatially average gradients—we directly weight activations by them and

²To process a novel video, we simply compute hotspots for each frame.

sum over channels. This is in contrast to GradCAM [64, 50] which produces maps that are useful for coarse object localization, but insufficient for interaction hotspots due to their low spatial resolution and diffuse global responses. Compared to simply applying GradCAM to an action recognition LSTM (Figure 3, top row), our model produces interaction hotspots that are significantly richer (bottom row). These differences are precisely due to both our anticipation distillation model trained jointly with the recognition model (Sec. 3.2), as well as the activation mapping strategy above. We provide a quantitative comparison in Sec. 4.

3.4. Training and Inference

During training (Figure 2, left), we generate embeddings $\{x_1, \dots, x_T\}$ for each frame of a video \mathcal{V} . These are passed through \mathbb{A} to generate the video embedding h_* , and then through a classifier to predict the afforded action label a . At the same time the inactive object image embedding x_I is computed and used to train the anticipation model to predict its *active* state \tilde{x}_I . The complete loss function for each training instance is:

$$\mathcal{L}(\mathcal{V}, \mathcal{I}, a) = \lambda_{cls} \mathcal{L}_{cls} + \lambda_{ant} \mathcal{L}_{ant} + \lambda_{aux} \mathcal{L}_{aux}, \quad (7)$$

where the λ terms control the weight of each component of the loss, and \mathcal{I} denotes the inactive object image.

For inference on an inactive test image (Figure 2, right), we first generate its image embedding x_I . Then, we hypothesize its *active* interaction embedding \tilde{x}_I , and use it to predict the afforded action scores. Finally, using Equation 6 we generate $|\mathcal{A}|$ heatmaps over x_I , one for each afforded action class. This stack of heatmaps are the *interaction hotspots*. Note that we produce activation maps for the original inactive image x_I , not for the hypothesized active output \tilde{x}_I , i.e., we propagate gradients *through* the anticipation network as well. Not doing so produces activation maps that are inconsistent with the input image, which hurts performance (see ablation study in Supp).

We stress that interaction hotspots are predictable even for unfamiliar objects. By training the afforded actions across object category boundaries, the system learns the general properties of appearance *and* interaction that characterize affordances. Hence, our approach can anticipate, for example, how an unfamiliar kitchen device might be used because it has learned how a variety of other objects operate. Similarly, heatmaps can be hallucinated for novel action-object pairs that have not been seen in training (e.g., “cut” using a spatula in Figure 4, bottom row).

Please see Supp. for complete implementation details.

4. Experiments

Our experiments on interaction hotspots explore their ability to describe affordances of objects, to generalize to

	Supervision Source	Type	N
EGOGAZE [27]	Recorded eye fixations	Weak	60k
SALIENCY [41, 6, 34]	Manual saliency labels	Weak	10k
OURS	Action, object labels	Weak	20k
IMG2HEATMAP	Manual affordance keypoints	Strong	20k
DEMO2VEC [12]	Manual affordance keypoints, action labels	Strong	20k

Table 1: Supervision source and type for all methods. Our method learns interaction hotspots *without* strong supervision like annotated segmentation/keypoints. N is the number of instances.

anticipate affordances of unfamiliar objects, and to encode functional similarities between object classes.

Datasets. We use two datasets:

- **OPRA** [12] contains videos of product reviews of appliances (e.g., refrigerators, coffee machines) collected from YouTube. Each instance is a short video demonstration \mathcal{V} of a product’s feature (e.g., pressing a button on a coffee machine) paired with a static image \mathcal{I} of the product, an interaction label a (e.g., “pressing”), and a manually created affordance heatmap \mathcal{M} (e.g., highlighting the button on the static image). There are $\sim 16k$ training instances of the form $(\mathcal{V}, \mathcal{I}, a, \mathcal{M})$, spanning 7 actions.
- **EPIC-Kitchens** [7] contains unscripted, egocentric videos of activities in a kitchen. Each clip \mathcal{V} is annotated with action and object labels a and o (e.g., cut tomato, open refrigerator) along with a set of bounding boxes \mathcal{B} (one per frame) for objects being interacted with. There are $\sim 40k$ training instances of the form $(\mathcal{V}, a, o, \mathcal{B})$, spanning 352 objects and 125 actions. We crowd-source annotations for ground-truth heatmaps \mathcal{M} resulting in 1.8k annotated instances over 20 action and 31 objects (see Supp. for details).

The two video datasets span diverse settings. OPRA has third person videos, where the person and the product being reviewed are clearly visible, and covers a small number of actions and products. EPIC-Kitchens has first-person videos of unscripted kitchen activities and a much larger vocabulary of actions and objects; the person is only partially visible when they manipulate an object. Together, they provide good variety and difficulty to evaluate the robustness of our model.³ For both datasets, our model uses only the action labels as supervision, and an *inactive* image for our anticipation loss \mathcal{L}_{ant} . We stress that (1) the annotated heatmap \mathcal{M} is used *only* for evaluation, and (2) the ground truth is well-aligned with our objective, since annotators were instructed to watch an interaction video clip to decide what regions to annotate for an object’s affordances.

³Other affordance segmentation datasets [37, 38] have minimal vocabulary overlap with OPRA/EPIC classes, and hence do not permit evaluation for our setting, since we learn from video.

While OPRA comes with an image \mathcal{I} of the *exact* product associated with each video instance, EPIC does not. Instead, we crop out inactive objects from frames using the provided bounding boxes \mathcal{B} , and randomly select one that matches the object class label in the video. To account for the appearance mismatch, in place of the L2 loss in Equation 5 we use a triplet loss, which uses “negatives” to ensure that inactive objects of the correct class can anticipate active features better than incorrect classes (see Supp. for details).

4.1. Interaction Hotspots as Grounded Affordances

In this section, we evaluate two things: 1) How well does our model learn object affordances? and 2) How well can it infer possible interactions for unfamiliar objects? For this, we train our model on video clips, and generate hotspot maps on *inactive* images where the object is at rest.

Baselines. We evaluate our model against several baselines and state-of-the-art models.

- **CENTER BIAS** produces a fixed Gaussian heatmap at the center of the image. This is a naive baseline to account for a possible center bias [6, 34, 41, 27].
- **LSTM+GRAD-CAM** uses an LSTM trained for action recognition with the same weak labels as our method, then applies standard Grad-CAM [50] to get heatmaps. It has no anticipation model.
- **SALIENCY** is a set of baselines that estimate the most salient regions in an image using models trained directly on saliency annotations/eye fixations: EGOGAZE [27], MLNET [6], DEEPGAZEII [34] and SALGAN [41]. We use the authors’ pretrained models.
- **DEMO2VEC** [12] is a supervised method that generates an affordance heatmap using context from a video demonstration of the interaction. We use the authors’ pre-computed heatmap predictions.
- **IMG2HEATMAP** is a supervised method that uses a fully convolutional encoder-decoder to predict the affordance heatmap for an image. It serves as a simplified version of DEMO2VEC that lacks video context during training.

The SALIENCY baselines capture a generic notion of spatial *importance*. They produce a single heatmap for an image, regardless of action class, and as such, are less expressive than our per-action-affordances. They are *weakly supervised* in that they are trained for a different task, albeit with strong supervision (heatmaps, gaze points) for that task. DEMO2VEC and IMG2HEATMAP are strongly supervised, and represent more traditional affordance learning techniques that learn affordances from manually labeled images [37, 46, 39, 10]. Table 1 summarizes the sources and types of supervision for all methods. Unlike other methods, our model uses only weak class labels during training.

		OPRA			EPIC			OPRA			EPIC		
		KLD ↓	SIM ↑	AUC-J ↑	KLD ↓	SIM ↑	AUC-J ↑	KLD ↓	SIM ↑	AUC-J ↑	KLD ↓	SIM ↑	AUC-J ↑
WS	CENTER BIAS	11.132	0.205	0.625	10.660	0.222	0.634	6.281	0.244	0.680	5.910	0.277	0.699
	LSTM+GRAD-CAM	8.573	0.209	0.620	6.470	0.257	0.626	5.405	0.259	0.644	4.508	0.255	0.664
	EGOGAZE [27]	2.428	0.245	0.646	2.241	0.273	0.614	2.083	0.278	0.694	1.974	0.298	0.673
	MLNET [6]	4.022	0.284	0.763	6.116	0.318	0.746	2.458	0.316	0.778	3.221	0.361	0.799
	DEEPGAZEII [34]	1.897	0.296	0.720	1.352	0.394	0.751	1.757	0.318	0.742	1.297	0.400	0.793
	SALGAN [41]	2.116	0.309	0.769	1.508	0.395	0.774	1.698	0.337	0.790	1.296	0.406	0.808
	OURS	1.427	0.362	0.806	1.258	0.404	0.785	1.381	0.374	0.826	1.249	0.405	0.817
SS	IMG2HEATMAP	1.473	0.355	0.821	1.400	0.359	0.794	1.431	0.362	0.820	1.466	0.353	0.770
	DEMO2VEC [12]	1.197	0.482	0.847	—	—	—	—	—	—	—	—	—

Grounded affordance prediction

Generalization to novel objects

Table 2: Interaction hotspot prediction results on OPRA and EPIC. **Left:** Our model outperforms other weakly supervised (WS) methods in all metrics, and approaches the performance of strongly supervised (SS) methods *without* the privilege of heatmap annotations during training. **Right:** Not only does our model generalize to new *instances*, but it also accurately infers interaction hotspots for novel object *categories* unseen during training. The proposed hotspots generalize on an object-function level. Values are averaged across three splits of object classes. (↑/↓ indicates higher/lower is better.) DEMO2VEC [12] is available only on OPRA and only for seen classes.

Grounded Affordance Prediction. First we compare the ground truth heatmaps for each interaction to our hotspots and the baselines’ heatmaps. We report error as KL-Divergence, following [12], as well as other metrics (SIM, AUC-J) from the saliency literature [2].

Table 2 (Left) summarizes the results. Our model outperforms all other weakly-supervised methods in all metrics across both datasets. These results highlight that our model can capture sophisticated interaction cues that describe more specialized notions of importance than saliency.

On OPRA, our model achieves relative improvements of up to 25% (KLD) compared to the strongest baseline, and it matches one of the strongly supervised baseline methods on two metrics. On EPIC, our model achieves relative improvements up to 7% (KLD). EPIC has a much larger, more granular action vocabulary, resulting in fewer and less spatially distinct hotspots. As a result, the baselines that produce redundant heatmaps for all actions artificially benefit on EPIC, though our results remain better.

The baselines have similar trends across datasets. Consistent with the examples in Figure 3, the LSTM+GRAD-CAM baseline in Table 2 demonstrates that simply training an action recognition model is clearly insufficient to learn affordances. Our anticipation model allows the system to bridge the (in)active gap between training video and test images, and is crucial for accuracy. All saliency methods perform worse than our model, despite that they may accidentally benefit from the fact that kitchen appliances have interaction regions designed to be visually salient (*e.g.*, buttons, handles). In contrast to our approach, none of the saliency baselines distinguish between affordances; they produce a single heatmap representing “important” salient points. To these methods, the blade of a knife is as important to the action “cutting” as it is to the action “holding”, and they are unable to explain objects with multiple affordances. IMG2HEATMAP and DEMO2VEC generate better affordance



Figure 4: Affordance heatmaps on inactive images. **Top:** Predicted affordance heatmaps for *hold*, *rotate*, *push* (red, green, blue) on OPRA. **Bottom row:** Predicted heatmaps for *cut*, *mix*, *turn-on* (red, green, blue), on EPIC. Our model highlights spatial affordances consistent with how people interact with the objects. Note that SALIENCY [41] produces only a single “importance” map (yellow). **Last column:** failure cases. Best viewed in color.

heatmaps, but at the cost of strong supervision. Our method actually approaches their accuracy without using any manual heatmaps for training.

Please see the Supp. file for an **ablation study** that further examines the contributions of each part of our model. In short, our class activation mapping strategy and propagating gradients all the way through the anticipation model are critical. All elements of the design play a role to achieve our full model’s best accuracy.

Figure 4 shows example heatmaps for inactive objects. Our model is able to highlight specific object regions that afford actions (*e.g.*, the knobs on the coffee machine as “rotatable” in column 1) after only watching videos of object

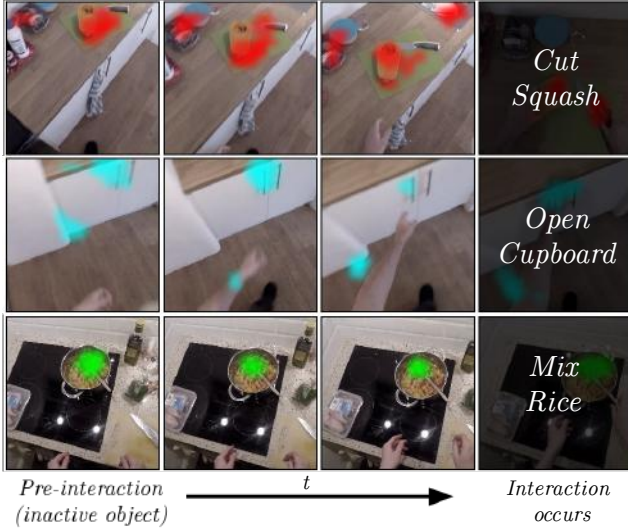


Figure 5: Interaction hotspots on EPIC videos of unseen object classes. Our model anticipates spatial interaction regions for inactive objects at rest (first column), *even before* the interaction happens. Critically, the object categories shown in this figure have *not* been seen during training; our model learns to generalize interaction hotspots. For example, there are no cupboards or squashes in the training videos, but our method anticipates how these objects would be opened and cut, respectively.

interactions. Weakly supervised SALIENCY methods highlight *all* salient object parts in a single map, regardless of the interaction in question. In contrast, our model highlights multiple distinct affordances for an object. To generate comparable heatmaps, DEMO2VEC requires annotated heatmaps for training *and* a set of video demonstrations during inference, whereas our model can hypothesize object functionality without these extra requirements.

Generalization to Novel Objects. Can interaction hotspots infer how *novel* object categories work? We next test if our model learns an *object-agnostic* representation for interaction—one that is not tied to object class. This is a useful property for open-world situations where unfamiliar objects may have to be interacted with to achieve a goal.

We divide the object categories \mathcal{O} into familiar and unfamiliar object categories $\mathcal{O} = \mathcal{O}_f \cup \mathcal{O}_u$; familiar ones are those seen with interactions in training video and unfamiliar ones are seen only during testing. We leave out 10/31 objects in EPIC and 9/26 objects in OPRA for our experiments, and divide our video train/test sets along these object splits. We train our model only on clips with the familiar objects from \mathcal{O}_f . If our model can successfully infer the heatmaps for novel, unseen objects, it will show that a general sense of object *function* is learned that is not strongly tied to object *identity*.

Table 2 (Right) shows the results. We see mostly similar trends as the previous section. On OPRA, our model out-

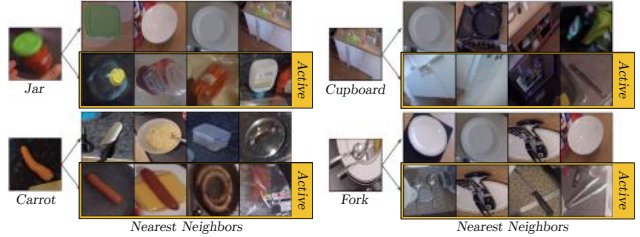


Figure 6: Inactive vs. active object embeddings. By hypothesizing potential interactions with objects, our model learns representations that capture functional similarities between objects across object classes, rather than purely appearance-based similarities.

performs all baselines in all metrics, and is able to infer the hotspot maps for unfamiliar object categories, despite never seeing them during training. On EPIC, our method remains the best weakly supervised method.

Qualitative results (Figure 5) support our numbers, showing our model applied to video clips from EPIC Kitchens, just before the action occurs. Our model—which was never trained on some objects (*e.g.*, cupboard, squash)—is able to anticipate characteristic spatial locations of interactions *before* the interaction occurs.

4.2. Interaction Hotspots for Functional Similarity

Finally, we show how our model encodes functional object similarities in its learned representation for objects. We compare the inactive object embedding space (standard ResNet features) to our predicted active embedding space (output of the anticipation model) by looking at nearest neighbor images in other object classes.

Figure 6 shows examples. Neighbors in the inactive object space (top branch) capture typical appearance-based visual similarities that are useful for object categorization—shapes, backgrounds, etc. In contrast, our active object space (bottom branch, yellow box) reorganizes the objects based on *how* they are interacted with. For example, fridges, cupboards, and microwaves, that are swung open in a characteristic way (top right); knives, spatulas, tongs, that are typically held at their handles (bottom right). Our model learns representations indicative of functional similarity between objects, despite the objects being visually distinct. See Supp. for a clustering visualization on all images.

5. Conclusion

We introduced a method to learn “interaction hotspot” maps—characteristic regions on objects that anticipate and explain object interactions—directly from watching videos of people naturally interacting with objects. Our experiments show that these hotspot maps explain object affordances better than other existing weakly supervised models and can generalize to anticipate affordances of unseen

objects. Furthermore, the representation learned by our model goes beyond appearance similarity to encode functional similarity. In future work, we plan to explore how hotspots might aid action anticipation and policy learning for robot-object interaction.

Acknowledgments: We would like to thank the authors of Demo2Vec [12] for their help with the OPRA dataset. We would also like to thank Marcus Rohrbach and Jitendra Malik for the helpful discussions.

References

- [1] J.-B. Alayrac, J. Sivic, I. Laptev, and S. Lacoste-Julien. Joint discovery of object states and manipulation actions. *ICCV*, 2017. 2
- [2] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand. What do different evaluation metrics tell us about saliency models? *TPAMI*, 2018. 7
- [3] C. Castellini, T. Tommasi, N. Noceti, F. Odone, and B. Caputo. Using object affordances to improve object recognition. *TAMD*, 2011. 1, 2
- [4] C.-Y. Chen and K. Grauman. Subjects and their objects: Localizing interactees for a person-centric view of importance. *IJCV*, 2016. 2
- [5] C.-Y. Chuang, J. Li, A. Torralba, and S. Fidler. Learning to act properly: Predicting and explaining affordances from images. *CVPR*, 2018. 2
- [6] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara. A Deep Multi-Level Network for Saliency Prediction. In *ICPR*, 2016. 6, 7
- [7] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, et al. Scaling egocentric vision: The epic-kitchens dataset. *ECCV*, 2018. 2, 6
- [8] D. Damen, T. Leelasawassuk, O. Haines, A. Calway, and W. W. Mayol-Cuevas. You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video. In *BMVC*, 2014. 2
- [9] V. Delaitre, D. F. Fouhey, I. Laptev, J. Sivic, A. Gupta, and A. A. Efros. Scene semantics from long-term observation of people. In *ECCV*, 2012. 2
- [10] T.-T. Do, A. Nguyen, I. Reid, D. G. Caldwell, and N. G. Tsagarakis. Affordancenet: An end-to-end deep learning approach for object affordance detection. *ICRA*, 2017. 1, 2, 3, 6
- [11] F. Ebert, C. Finn, A. X. Lee, and S. Levine. Self-supervised visual planning with temporal skip connections. *CORL*, 2017. 3
- [12] K. Fang, T.-L. Wu, D. Yang, S. Savarese, and J. J. Lim. Demo2vec: Reasoning object affordances from online videos. In *CVPR*, 2018. 1, 2, 3, 6, 7, 9, 11, 12, 14
- [13] C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *NeurIPS*, 2016. 3
- [14] C. Finn and S. Levine. Deep visual foresight for planning robot motion. In *ICRA*, 2017. 3
- [15] D. F. Fouhey, V. Delaitre, A. Gupta, A. A. Efros, I. Laptev, and J. Sivic. People watching: Human actions as a cue for single view geometry. *IJCV*, 2014. 2
- [16] A. Furnari, S. Battiato, K. Grauman, and G. M. Farinella. Next-active-object prediction from egocentric videos. *JVCIR*, 2017. 3
- [17] J. J. Gibson. *The ecological approach to visual perception: classic edition*. Psychology Press, 1979. 1, 2
- [18] H. Grabner, J. Gall, and L. Van Gool. What makes a chair a chair? In *CVPR*, 2011. 1, 2
- [19] A. Gupta and L. S. Davis. Objects in action: An approach for combining action understanding and object perception. In *CVPR*, 2007. 2
- [20] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *TPAMI*, 2009. 2
- [21] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert. From 3d scene geometry to human workspace. In *CVPR*, 2011. 2
- [22] M. Hassanin, S. Khan, and M. Tahtali. Visual affordance and function understanding: A survey. *arXiv preprint arXiv:1807.06775*, 2018. 1, 2
- [23] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 12
- [24] T. Hermans, J. M. Rehg, and A. Bobick. Affordance prediction via learned object attributes. In *ICRA: Workshop on Semantic Perception, Mapping, and Exploration*, 2011. 1
- [25] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 1997. 3
- [26] D.-A. Huang and K. M. Kitani. Action-reaction: Forecasting the dynamics of human interaction. In *ECCV*, 2014. 3
- [27] Y. Huang, M. Cai, Z. Li, and Y. Sato. Predicting gaze in egocentric video by learning task-dependent attention transition. *ECCV*, 2018. 6, 7
- [28] D. Jayaraman, F. Ebert, A. A. Efros, and S. Levine. Time-agnostic prediction: Predicting predictable video frames. *ICLR*, 2019. 3
- [29] H. Kjellström, J. Romero, and D. Kragić. Visual object-action recognition: Inferring object affordances from human demonstration. *CVIU*, 2011. 2
- [30] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *IJR*, 2013. 2
- [31] H. S. Koppula and A. Saxena. Physically grounded spatio-temporal object affordances. In *ECCV*, 2014. 1, 2
- [32] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. *TPAMI*, 2016. 2, 3
- [33] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 12
- [34] M. Kümmerer, T. S. Wallis, and M. Bethge. Deepgaze ii: Reading fixations from deep features trained on object recognition. *arXiv preprint arXiv:1610.01563*, 2016. 6, 7
- [35] X. Liang, L. Lee, W. Dai, and E. P. Xing. Dual motion gan for future-flow embedded video prediction. In *ICCV*, 2017. 3

- [36] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *ICLR*, 2016. 3
- [37] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos. Affordance detection of tool parts from geometric features. In *ICRA*, 2015. 1, 2, 3, 6
- [38] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis. Detecting object affordances with convolutional neural networks. In *IROS*, 2016. 1, 6
- [39] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis. Object-based affordances detection with convolutional neural networks and dense conditional random fields. In *IROS*, 2017. 1, 2, 3, 6
- [40] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh. Action-conditional video prediction using deep networks in atari games. In *NeurIPS*, 2015. 3
- [41] J. Pan, C. C. Ferrer, K. McGuinness, N. E. O'Connor, J. Torres, E. Sayrol, and X. Giro-i Nieto. Salgan: Visual saliency prediction with generative adversarial networks. *arXiv preprint arXiv:1701.01081*, 2017. 6, 7
- [42] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *CVPR*, 2012. 4
- [43] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014. 3
- [44] N. Rhinehart and K. Kitani. First-person activity forecasting from video with online inverse reinforcement learning. *TPAMI*, 2018. 3
- [45] N. Rhinehart and K. M. Kitani. Learning action maps of large environments via first-person vision. In *CVPR*, 2016. 2
- [46] A. Roy and S. Todorovic. A multi-scale cnn for affordance segmentation in rgb images. In *ECCV*, 2016. 1, 2, 3, 6
- [47] M. Savva, A. X. Chang, P. Hanrahan, M. Fisher, and M. Nießner. Scenegrok: Inferring action maps in 3d environments. *TOG*, 2014. 2
- [48] J. Sawatzky and J. Gall. Adaptive binarization for weakly supervised affordance segmentation. *ICCV: Workshop on Assistive Computer Vision and Robotics*, 2017. 3
- [49] J. Sawatzky, A. Srikantha, and J. Gall. Weakly supervised affordance detection. In *CVPR*, 2017. 1, 2, 3
- [50] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017. 5, 6
- [51] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *ICLR: Workshop*, 2015. 5
- [52] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015. 3
- [53] M. Stark, P. Lies, M. Zillich, J. Wyatt, and B. Schiele. Functional object class detection based on learned affordance cues. In *ICVS*, 2008. 2
- [54] S. Thermos, G. T. Papadopoulos, P. Daras, and G. Potamianos. Deep affordance-grounded sensorimotor object recognition. *CVPR*, 2017. 2
- [55] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee. Learning to generate long-term future via hierarchical prediction. In *ICML*, 2017. 3
- [56] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating the future by watching unlabeled video. *CVPR*, 2016. 3
- [57] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *NeurIPS*, 2016. 3
- [58] C. Vondrick and A. Torralba. Generating the future with adversarial transformers. In *CVPR*, 2017. 3
- [59] J. Walker, K. Marino, A. Gupta, and M. Hebert. The pose knows: Video forecasting by generating pose futures. In *ICCV*, 2017. 3
- [60] X. Wang, R. Girdhar, and A. Gupta. Binge watching: Scaling affordance learning from sitcoms. In *CVPR*, 2017. 2
- [61] T. Xue, J. Wu, K. Bouman, and B. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *NeurIPS*, 2016. 3
- [62] B. Yao and L. Fei-Fei. Grouplet: A structured image representation for recognizing human and object interactions. In *CVPR*, 2010. 2
- [63] B. Yao, J. Ma, and L. Fei-Fei. Discovering object functionality. In *ICCV*, 2013. 2
- [64] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 5
- [65] Y. Zhou, B. Ni, R. Hong, X. Yang, and Q. Tian. Cascaded interactional targeting network for egocentric video analysis. In *CVPR*, 2016. 2
- [66] Y. Zhu, C. Jiang, Y. Zhao, D. Terzopoulos, and S.-C. Zhu. Inferring forces and learning human utilities from videos. In *CVPR*, 2016. 1, 2
- [67] Y. Zhu, Y. Zhao, and S. Chun Zhu. Understanding tools: Task-oriented object modeling, learning and recognition. In *CVPR*, 2015. 1, 2

Supplementary Material

This section contains supplementary material to support the main paper text. The contents include:

- (§S1) A video demonstrating our method on clips of human-object interaction.
- (§S2) Ablation study of our model components in Section 3.2 and Section 3.3 of the main paper.
- (§S3) EPIC Kitchens data annotation details.
- (§S4) Details about the anticipation loss \mathcal{L}_{ant} for EPIC Kitchens described in Section 4 (Datasets).
- (§S5) Implementation details for our model and experiments in Section 4.1.
- (§S6) Architecture details for the IMG2HEATMAP model introduced in Section 4.1 (Baselines)
- (§S7) Additional details about the evaluation protocol for our experiments in Section 4.1.
- (§S8) More examples of hotspot predictions on OPRA and EPIC to supplement Figure 4 in the main paper.
- (§S9) Clustering visualizations to accompany the results in Section 4.2.

S1. Interaction hotspots generated on video clips

We demonstrate our method on videos from EPIC by computing hotspots for each frame of a video clip. Note that the OPRA test set consists only of static images (Table 2 in main paper), as following the evaluation protocol in [12]. The video can be found on the [project page](#). Our model is trained on all actions, but we show hotspots for 5 frequent actions (cut, mix, adjust, open and wash) for clarity. Our model is able to derive hotspot maps on inactive objects *before* the interaction takes place. During this time, the objects are at rest, and are not being interacted with, yet our model can anticipate how they could be interacted with. In contrast to prior weakly supervised saliency methods, our model generates distinct maps for each action that aligns with object function. Some failure modes include when our model is presented rare objects/actions or unfamiliar viewpoints, for which our model produces diffused or noisy heatmaps.

S2. Ablation study

As noted in the main paper (Section 4.1), we study the effect of each proposed component in Section 3.2 and Section 3.3 on the performance of our model. Table S1 shows how they contribute towards more affordance aware activation maps. Specifically, we see that increasing the backbone resolution to N=28 increases AUC-J from 0.707 to 0.766.

	OPRA			EPIC		
	KLD ↓	SIM ↑	AUC-J ↑	KLD ↓	SIM ↑	AUC-J ↑
OURS (BASIC)	1.561	0.349	0.707	1.342	0.396	0.714
+ RESOLUTION	1.492	0.352	0.766	1.343	0.395	0.731
+ L2 POOL	1.489	0.349	0.770	1.361	0.385	0.727
+ ANTICIPATION	1.427	0.362	0.806	1.258	0.404	0.785

Table S1: Ablation study of model enhancements. Using a larger backbone feature resolution, L2-pooling, and the anticipation module improves our model performance, and are essential for well localized hotspot maps.

Using L2-pooling to ensure that spatial locations produce gradients as a function of their activation magnitudes increases this to 0.770, and using our anticipation model (Section 3.2 of the main paper) further increases AUC-J to **0.806**. Note that within our model, hotspots *could* be derived at the output hypothesized image, corresponding to hotspots on a regular video frame, but this does not align with the static images, causing a drop in performance (0.806 vs. 0.723 AUC-J). Propagating gradients back through the anticipation module produces correctly aligned gradients. More generally, we observe consistent improvement for our components across all metrics on both OPRA and EPIC.

S3. Data collection setup for EPIC-Kitchens annotations

As mentioned in Section 4 (Datasets), we collect annotations for interaction keypoints on EPIC Kitchens in order to quantitatively evaluate our method in parallel to the OPRA dataset (where annotations are available). We note that these annotations are collected purely for evaluation, and are not used for training our model. We select the 20 most frequent verbs, and select 31 nouns that afford these interactions. The list of verbs and nouns can be found in Table S2. To select instances for annotation, we first identify frames in the EPIC Kitchen videos which contain the object, and crop out the object using the provided bounding box annotations. Most of these object crops are *active* *i.e.* the objects are not at rest and/or they are being actively manipulated. We discard instances where hands are present in the crop using an off the shelf hand detection model, and manually select for *inactive* images from the remaining object crops. These images are then annotated.

We crowdsource these annotations using the Amazon Mechanical Turk platform. Following [12], our annotation setup is as follows. A user is asked to watch a short video (2-5s) of an object interaction (eg. person cutting carrot), and place 1-2 points on an image of the object for where the interaction is performed. We solicited 5 annotations for the same image (from unique users) to account for inter-annotator variance. Overall, we collected 19800 responses from 613 workers for our task, resulting in 1871 annotated

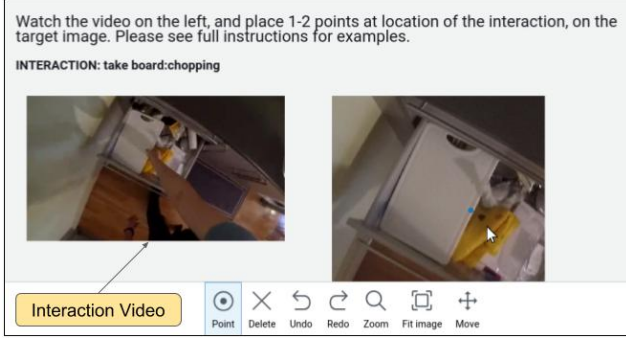


Figure S1: Interface for collecting annotations on EPIC. Users are asked to watch a video containing an interaction, and mark keypoints at the location of the interaction.

verbs	take, put, open, close, wash, cut, mix, pour, throw, move, remove, dry, turn-on, turn, shake, turn-off, peel, adjust, empty, scoop
nouns	board:chopping, bottle, bowl, box, carrot, colander, container, cup, cupboard, dishwasher, drawer, fork, fridge, hob, jar, kettle, knife, ladle, lid, liquid:washing, microwave, pan, peeler:potato, plate, sausage, scissors, spatula, spoon, tap, tongs, tray

Table S2: Verbs and nouns annotated for EPIC.

(image, verb) pairs. Our annotation interface is shown in Figure S1.

Finally, following [12], we convert these annotations into a heatmap by centering a gaussian distribution over each point. We use this heatmap as our ground truth \mathcal{M} described in Section 4 (Datasets). Some examples of these collected annotations and their correspondingly derived heatmaps are shown in Figure S2.

Compared to asking annotators to label static images with affordance labels (e.g., label "openable"), annotations collected by watching videos and then placing points is well-aligned with our objective of learning fine-grained object interaction. The annotations are better localized and are grounded in real human demonstration, making them meaningful for evaluation.

S4. Anticipation loss for EPIC-Kitchens

As mentioned in Section 3.2 and Section 3.3, we require inactive images to train the anticipation model. For OPRA, these are the catalog photos of the object provided with each video. In EPIC, we crop out inactive objects from frames using the provided bounding boxes \mathcal{B} , and select the inactive image that has the same class label as the object in the video. Unlike OPRA, these images may be visually different from the object in the video, preventing us from using the L2 loss directly. Instead, we use a triplet loss for the

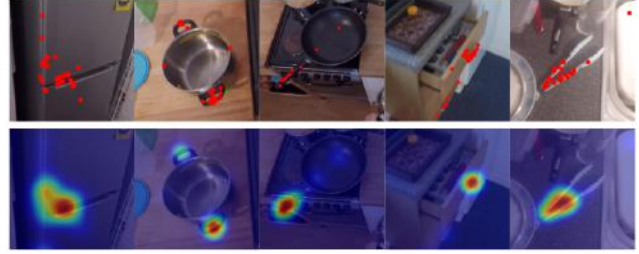


Figure S2: Top: Some example annotations provided as key-points on EPIC object crops **Bottom:** The resulting heatmaps derived from these annotations, to be used as ground truth for evaluation.

anticipation loss term as follows:

$$\mathcal{L}'_{ant}(\mathcal{V}, x_I, x_{I'}) = \max [0, d(P(x_{t^*}), P(\tilde{x}_I)) - d(P(x_{t^*}), P(\tilde{x}_{I'})) + M],$$

where x_I and $x_{I'}$ represent inactive image features with the correct and incorrect object class respectively. d denotes Euclidean distance, and M is the margin value. We normalize the inputs before computing the triplet loss, thus we keep the margin value fixed at 0.5. This term ensures that inactive objects of the correct class can anticipate active features better than incorrect classes, and is less sensitive to appearance mismatches compared to Equation 5 in the main paper.

S5. Implementation details

We provide implementation and training details for our experiments in Section 4. For all experiments, we use an ImageNet [33] pretrained ResNet-50 [23] modified for higher output resolution. To increase the output dimension from $n = 7$ to $n = 28$, we set the spatial stride of res_4 and res_5 to 1 (instead of 2), and use dilation of 2 (res_4) and 4 (res_5) for its filters to preserve the original scale. For \mathcal{F}_{ant} we use 2 sets of (conv-bn-relu) blocks, with 3x3 convolutions, maintaining the channel dimension $d = 2048$. We use a single layer LSTM (hidden size 2048) as \mathbb{A} , and train using chunks of 16 frames at a time. For our combined loss (Equation 7 in the main paper), we set $\lambda_{cls} = \lambda_{aux} = 1$ and $\lambda_{ant} = 0.1$ based on validation experiments. Our models are implemented in PyTorch. Adam with learning rate $1e-4$, weight decay $5e-4$ and batch size 128 is used to optimize the models parameters.

S6. Architecture details for supervised baselines

The IMG2HEATMAP model in Section 4.1 is a fully convolutional encoder-decoder to predict the affordance heatmap for an image. The encoder is an ImageNet pretrained VGG16 backbone (up to conv5), resulting in an encoded feature with 512 channels and spatial extent 7. This

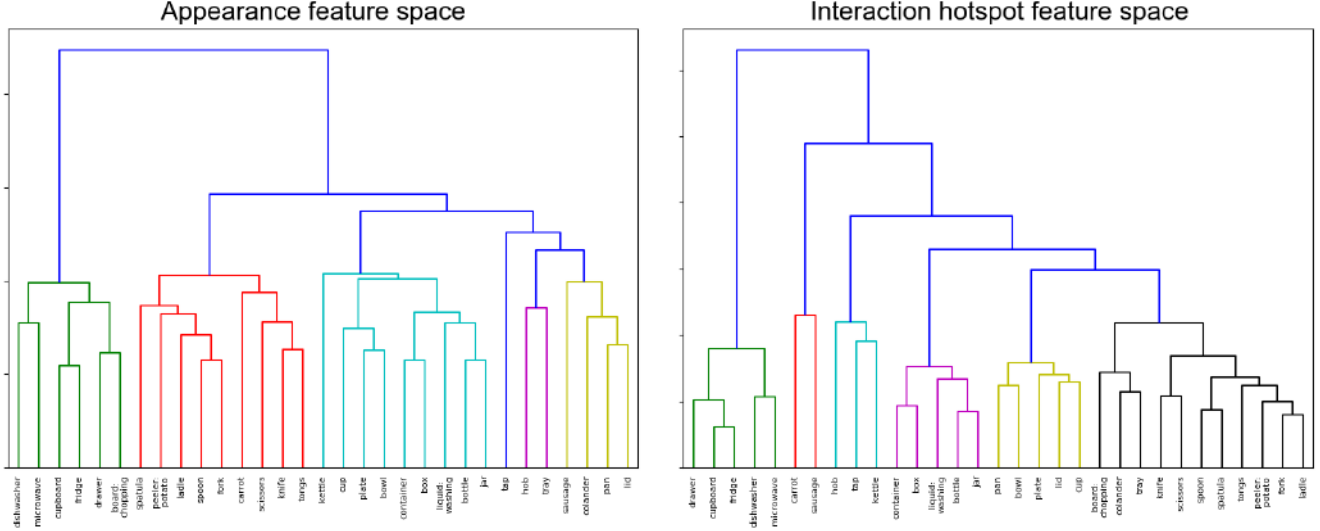


Figure S3: Clustering of the average object embedding in the appearance vs. interaction hotspot feature space. Appearance features capture similarities in shapes and textures (knife, tongs) and object co-occurrence (pan, lid; cup, kettle). In contrast, our interaction hotspot features encode similarities that are aligned with object function and use (cupboard, microwave, fridge - characteristically swung open; carrot, sausage - cut and held similarly). Similarity in this space refers to L2 distance between average object embeddings.

feature is passed through a decoder with an architecture mirroring the backbone, where the max-pooling operations are replaced with bilinear upsampling operations. This results in an output of the same dimension as the input, and as many channels as the number of actions. The output of this network is fed through a sigmoid operator and reconstruction loss against the ground truth affordance heatmap is calculated using binary cross-entropy.

S7. Evaluation protocol for grounded affordance prediction

As discussed in Section 4.1, the heatmaps generated by our model and the baselines are evaluated against the manually annotated ground truth heatmaps provided in the OPRA dataset and collected on EPIC (results in Table 2). For a single action (e.g. “press” a button), the ground truth heatmaps may occur distributed across several instances (e.g. different clips of people pressing different buttons on the same object). We simply take the union of all these heatmaps as our target affordance heatmap for the action. For evaluation, this leaves us with 1042 (image, action) pairs in OPRA, and 571 (image, action) pairs in EPIC. For AUC-J, we binarize heatmaps using a threshold of 0.5 for evaluation.

S8. Additional examples of generated hotspot maps

We provide more examples of our hotspot maps to accompany our results in the main paper. Figure S4 and Figure S5 contains more examples of these on OPRA and EPIC

respectively to supplement our results in Figure 4 in the main paper. Unlike the baselines, our model highlights multiple distinct affordances for an object and does so without heatmap annotations during training. The last 4 images in each figure show some failure cases where our model is unable to produce heatmaps for small or unfamiliar objects.

S9. Clustering visualization for appearance vs. our interaction hotspot features

We show the full clustering of objects in the appearance vs. interaction hotspot feature space to supplement the nearest neighbor visualizations presented in Section 4.2 of the main paper. Each object is represented by a vector obtained by averaging the embeddings of all instances for that specific object class. The resultant *average object representations* for all classes are then clustered using agglomerative hierarchical clustering. L2 distance in this space represents average similarity between object classes.

Figure S3 shows how our learned representation groups together objects related by their function and interaction modality, more so than the original appearance-based visual representation. Appearance features capture similarities in shapes and textures (knife, tongs) and object co-occurrence (pan, lid; cup, kettle). In contrast our representation encodes object function. Cupboards, microwaves, fridges that are characteristically swung opened; knives and scissors that afford the same cutting action; carrots, sausages that are cut and held in the same manner, are clustered together.



Figure S4: Generated affordance heatmaps on inactive images from OPRA. Our interaction hotspot maps show holdable, rotatable, and pushable regions (in red, green, and blue respectively). Saliency heatmaps do not discriminate between interactions and produce a single heatmap shown in yellow. Recall that the DEMO2VEC approach [12] is strongly supervised, whereas our approach is weakly supervised. Some failure cases due to small or unfamiliar objects can be seen in the last 4 examples.

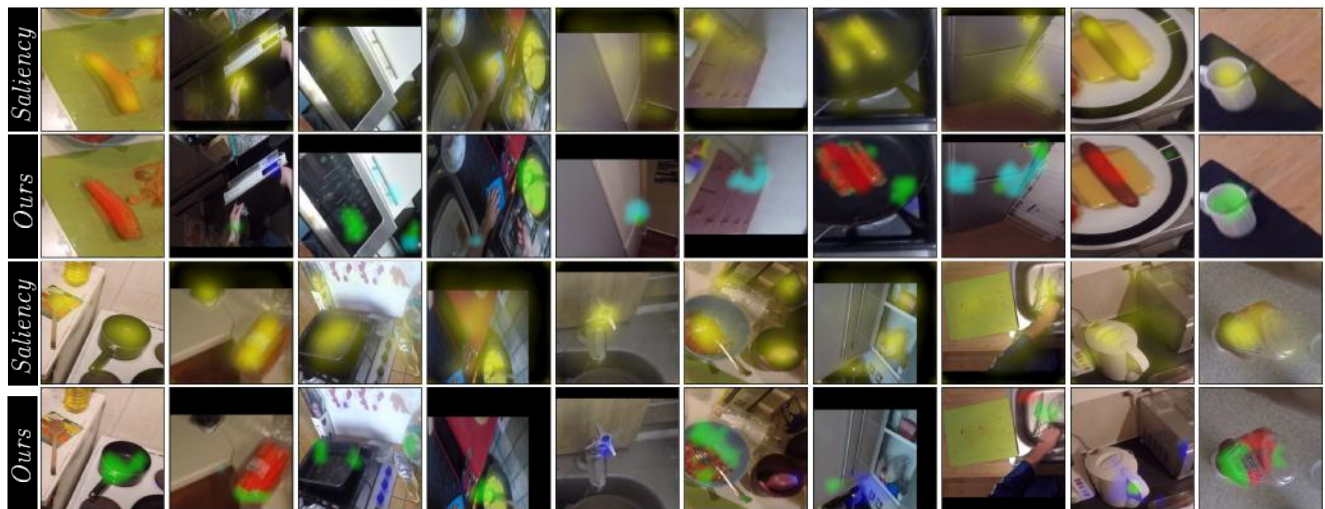


Figure S5: Generated interaction hotspot maps on inactive images from EPIC-Kitchens. Our interaction hotspot maps show cuttable, mixable, adjustable, and openable regions (in red, green, blue, and cyan, respectively). Failure cases can be seen in the last 4 examples.