Interaction Hotspots From Video (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 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. 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 without the anticipation module, the network does not have access to the static images. DIRECT CLS is a CNN (not LSTM) variant of our model that is trained to directly predict affordance classes from individual images. It has access to all the data our model uses during training—video frames and the inactive images used by our anticipation module—but it overall underperforms our method. Note that within our model, hotspots *could* be 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.

	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**
DIRECT CLS	1.606	0.343	0.682	1.263	**0.408**	0.767

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.

S3. Data Collection Setup for EPIC-Kitchens Annotations

As mentioned in Section 4 (Datasets), we collect annotations for interaction keypoints on EPIC Kitchens to quantitatively evaluate our method in parallel to OPRA (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 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 detector, and manually select *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

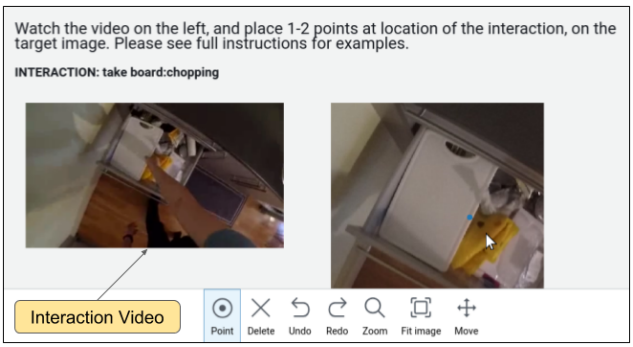

Figure S1: Interface for collecting annotations on EPIC. Users are asked to watch a video containing an interaction, and mark keypoints at the location of the interaction.

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.

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 (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

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

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 anticipation loss term as follows:

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

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. Supervised Baseline Architecture Details

The IMG2HEATMAP model in Section 4.1 is a fully convolutional encoder-decoder to predict the affordance

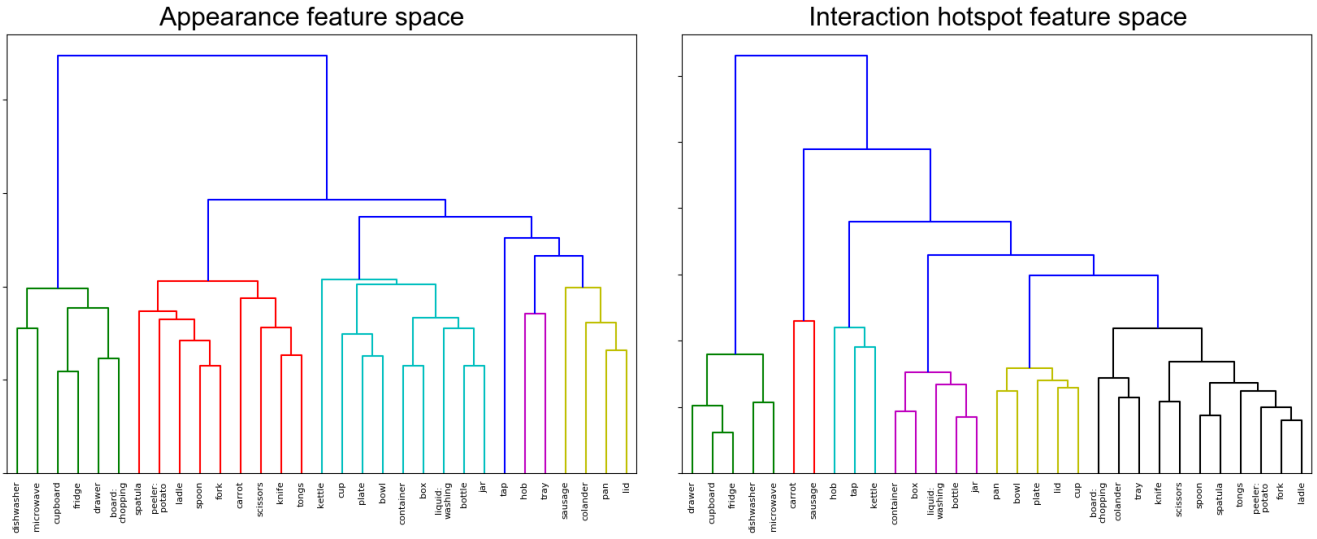

Figure S3: Clustering of the average object embedding in the appearance vs. interaction hotspot feature space. Appearance features capture similarities in shapes and textures (knife, tongs) and object co-occurrence (pan, lid; cup, kettle). In contrast, our interaction hotspot features encode similarities 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.

heatmap for an image. The encoder is an ImageNet pre-trained VGG16 backbone (up to conv5), resulting in an encoded feature with 512 channels and spatial extent 7. This 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 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.

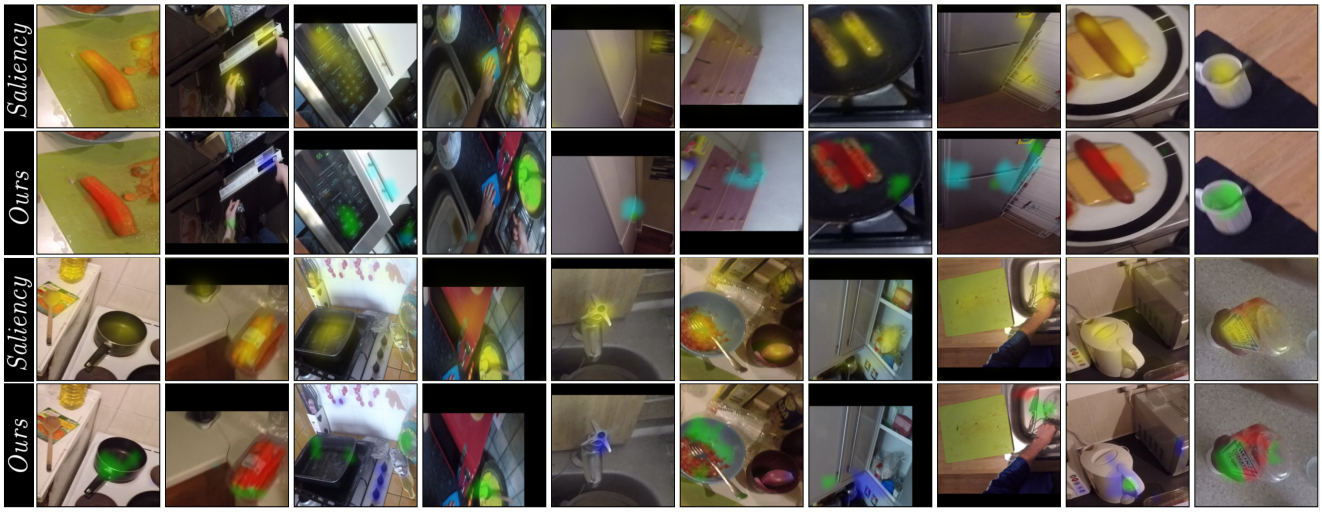

Figure S5: Generated interaction hotspot maps on inactive images from EPIC-Kitchens. Our interaction hotspot maps show cuttable, mixable, adjustable, and openable regions (in red, green, blue, and cyan, respectively). Failure cases can be seen in the last 4 examples.