

# 2HandedAfforder: Learning Precise Actionable Bimanual Affordances from Human Videos

## Supplementary Material

### 7. Additional Filtering and Augmentation Steps

Between each of the major steps of the affordance extraction pipeline different filtering steps were applied to clean up the data. After calculating the intersection of the completed mask and the hand mask erosion and dilation steps were applied to remove scattered mask pixels and fill gaps to leave only one connected affordance mask. Furthermore, inconsistencies and inaccuracies within the data were detected and the data points were deleted, e.g. the calculated affordance masks are empty, the action is classified as bimanual but only one affordance mask is provided. Lastly, we remove datapoints with narrations that are too vague or do not describe an affordance by blacklisting some expressions, e.g. ‘throw **something** into the bin’ or ‘**look** at pan’. We augment the data by flipping all of the actions horizontally, essentially doubling the size of the dataset. By doing that we even out the ratio of left-handed and right-handed actions. Afterwards, we apply common augmenting strategies also used by Goyal et al. [12], i.e., color jittering (randomly changing the brightness, contrast, saturation and hue of the inpainted frame) and cropping.

### 8. 2HANDS Dataset

Each data point of 2HANDS consists of an inpainted frame, two affordance masks where one of them is left empty if it is a unimanual action and the narration. We also provide additional information such as the object masks and object names if needed. In the end, the proposed dataset 2HANDS consists of over 278k datapoints from 25 different kitchen environments from the EPIC-KITCHENS dataset. An overview of the dataset can be found in Table 4.1. This dataset was used to train the models.

	Amount
<b>Left Handed</b>	76,278
<b>Right Handed</b>	76,278
<b>Symmetric</b>	51,684
<b>Asymmetric</b>	73,920
<b>Total</b>	278,160
<b>No. Kitchen Environments</b>	25
<b>No. Videos</b>	47
<b>No. Object Classes</b>	160
<b>No. Verb Classes</b>	73

Table 3. Overview of the dataset

We collected affordance masks for 160 different object categories and 73 verb class.

The object classes:

alarm, almond, aubergine, bag, banana, basil, bean:green, beer, bin, board:chopping, book, bottle, bowl, box, bread, broccoli, brush, butter, button, can, candle, cap, caper, carrot, chair, cheese, cherry, chicken, chilli, choi:pak, chopstick, cinnamon, cloth, clothes, coffee, colander, container, cooker:slow, cork, corn, cover, cucumber, cumin, cup, cupboard, cutlery, cutter:pizza, dishwasher, dough, drawer, fan:extractor, filter, fish, flour, food, fork, fridge, garlic, ginger, glass, glove, grater, hand, heat, heater, hob, holder, ice, jar, jug, juicer, kettle, knife, knob, label, ladle, leaf, leek, lemon, lettuce, lid, light, lighter, liquid:washing, machine:sous:vide, machine:washing, maker:coffee, mat, meat, microwave, milk, mixture, mushroom, napkin, noodle, nut, oil, onion, opener:bottle, oven, package, pan, pan:dust, paper, paste, peach, peeler:potato, pepper, phone, pin:rolling, pith, pizza, plate, plug, pork, pot, potato, powder:washing, processor:food, rack:drying, rest, rice, roll, rubbish, salt, sauce, sausage, scissors, shell:egg, sink, skin, soda, spatula, sponge, spoon, sprout, stalk, stock, syrup, tap, toaster, tofu, tomato, tongs, top, towel, towel:kitchen, tray, utensil, vegetable, vinegar, wall, water, whetstone, window, wine, wire, wrap, wrap:plastic, yoghurt

The verb classes:

add, adjust, apply, attach, break, brush, carry, check, choose, close, coat, cook, crush, cut, divide, drink, dry, empty, fill, filter, flatten, flip, form, gather, hold, increase, insert, knead, lift, lower, mix, move, open, pat, peel, pour, press, pull, put, remove, rip, roll, rub, scoop, scrape, screw, scrub, season, serve, set, shake, sharpen, slide, soak, sort, spray, sprinkle, squeeze, stab, stretch, take, throw, turn, turn-down, turn-off, turn-on, uncover, unroll, unscrew, unwrap, use, wash, wrap

*The full dataset and codebase will be released at [sites.google.com/view/2handedafforder](https://sites.google.com/view/2handedafforder).*

### 9. Additional Qualitative Results



Figure 7. Failure cases. In both cases the model is undecided of what to do. In the left example it predicts affordance regions at different objects that are also not related to the task, i.e., the spatula and the knife. In the right example the model predicts a bimanual action to pick up the bowl and predicts affordance regions at multiple bowls even though only one bowl is supposed to be picked up.

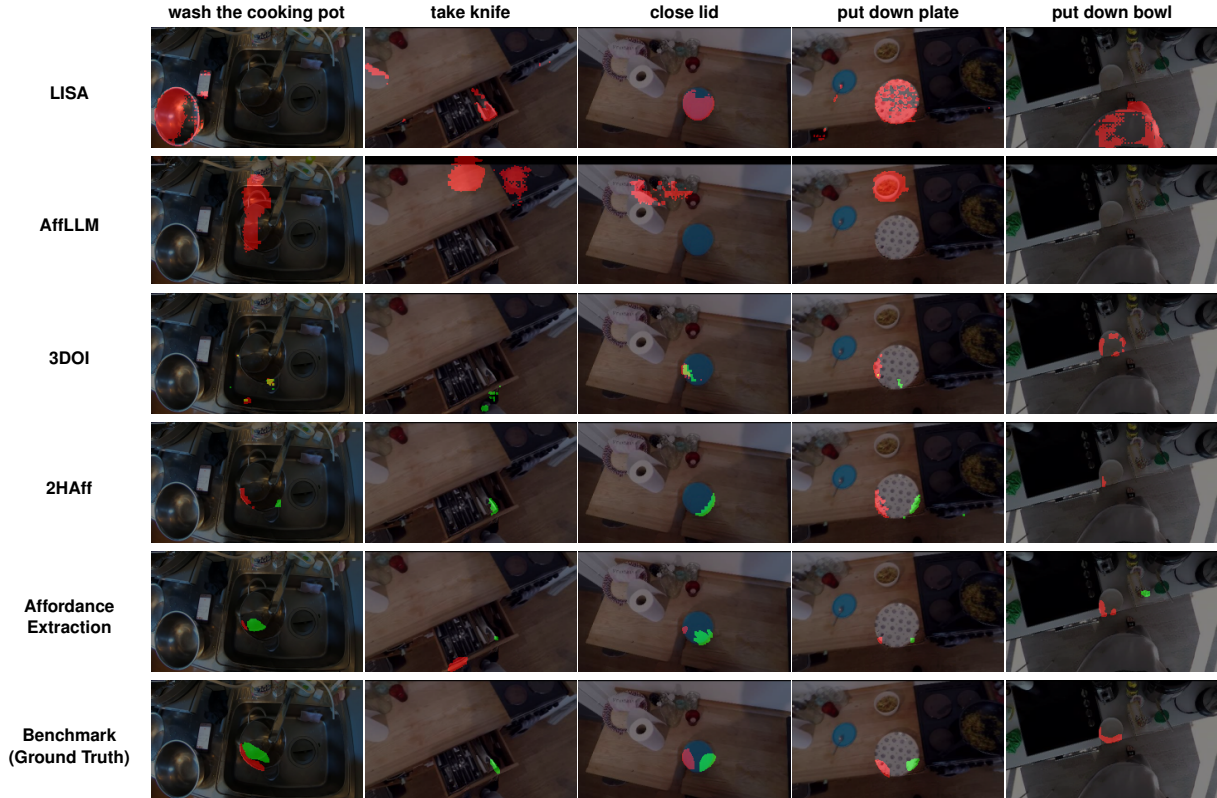


Figure 8. Additional qualitative results showing the performance of our proposed model compared to different baselines and the ground truth.

## 10. ActAffordance Annotation Procedure

For annotating the images for the ActAffordance Benchmark, we used TORAS [19]. We asked 10 human annotators to highlight all possible interaction regions of the target objects in the image where the hands were already removed with respect to the underlying task, i.e. the narration. This

annotation was done for both the left and right hands. Additionally, annotators also had access to the original image to see how the hands interacted with the objects in the scene.

## 11. Real robot experiments

A successful example for an affordance prediction as well as the corresponding masks from LangSAM are visualized in Figure 9.



Figure 9. The affordance detection of our method detects precise affordance regions (left) for the pot and the spatula that can be used to successfully perform the task of stirring within the pot. The right image shows the mask outputs by LangSAM. The text prompts used for this prediction were “wooden spatula” and “cooking pot” for LangSAM and “stir vegetables” for 2HandedAfforder.

Figure 10 shows the robot performing the task of ‘stirring vegetables’. The first example illustrates an unsuccessful attempt where the robot, relying on plain object segmentation, attempts to stir within the pot while holding the spatula too close to the middle. This suboptimal grip prevents the robot from reaching into the pot, making it impossible for it to complete the task of stirring the vegetables properly. The second example demonstrates an improvement, as the robot uses our affordance detection method to identify a better grasping region. However, it employs only one arm instead of two, leading to an unintended side effect since the pot is not stabilized. The stirring motion causes it to move, making the task more difficult.

In the final and most successful example, the robot fully utilizes both affordance regions detected by 2HandedAfforder. Here, the left end effector grasps the spatula closer to its edge while the right end effector holds the pot securely in place. This configuration enables a stable and effective

stirring motion, demonstrating the advantages of incorporating our affordance predictions in bimanual robotic manipulation tasks.



Figure 10. Demonstration how different affordance detections determine the success in performing a specific task.

## 12. Qualitative Analysis of Mask Completion Approaches

We developed and evaluated two mask completion approaches, i.e., an image reconstruction (IR) based and video segmentation (VS) based approach. The IR based approach uses the image inpainting model MI-GAN [46] to inpaint the missing regions of the mask using the hand mask as inpainting region. The VS based approach creates an image sequence out of the original image and the image with the hands removed. The object mask for the original image is then propagated to the inpainted image using SAM2 [42] to create the completed version of the mask. The evaluation of these approaches was conducted qualitatively, and some examples can be seen in Fig. 11. It is clearly visible that the VS based approach performs better on average than the IR based approach. Generally, the VS based approach provides more accurate results, see row 1 and 2, and it does not detect affordances at regions where the object was not reconstructed properly (row 3 and 4). This can be explained intuitively for two reasons: The IR based approach has no information about the underlying RGB image and only processes the object mask itself which is binary by nature. So the image reconstruction model focuses on simple principles such as the continuation of lines and shapes. This leads to the reconstruction of what we call ‘ghost handles’ where

the image reconstruction model still predicts the existence of a handle or object part in general even though the object was not reconstructed successfully. This also reduces the accuracy of the IR based method. The VS based approach, however, has the information about the inpainted image and thus will only complete object parts that are properly inpainted. So, it naturally filters out data points where the object was not inpainted properly since the completed mask will not intersect the hand mask. Thus, it will always be more accurate and never predict 'ghost handles'. There are only a couple of examples where the IR based approach performs better than the VS based approach (row 5). Hence, we decided to use the VS based approach for the creation of 2HANDS.



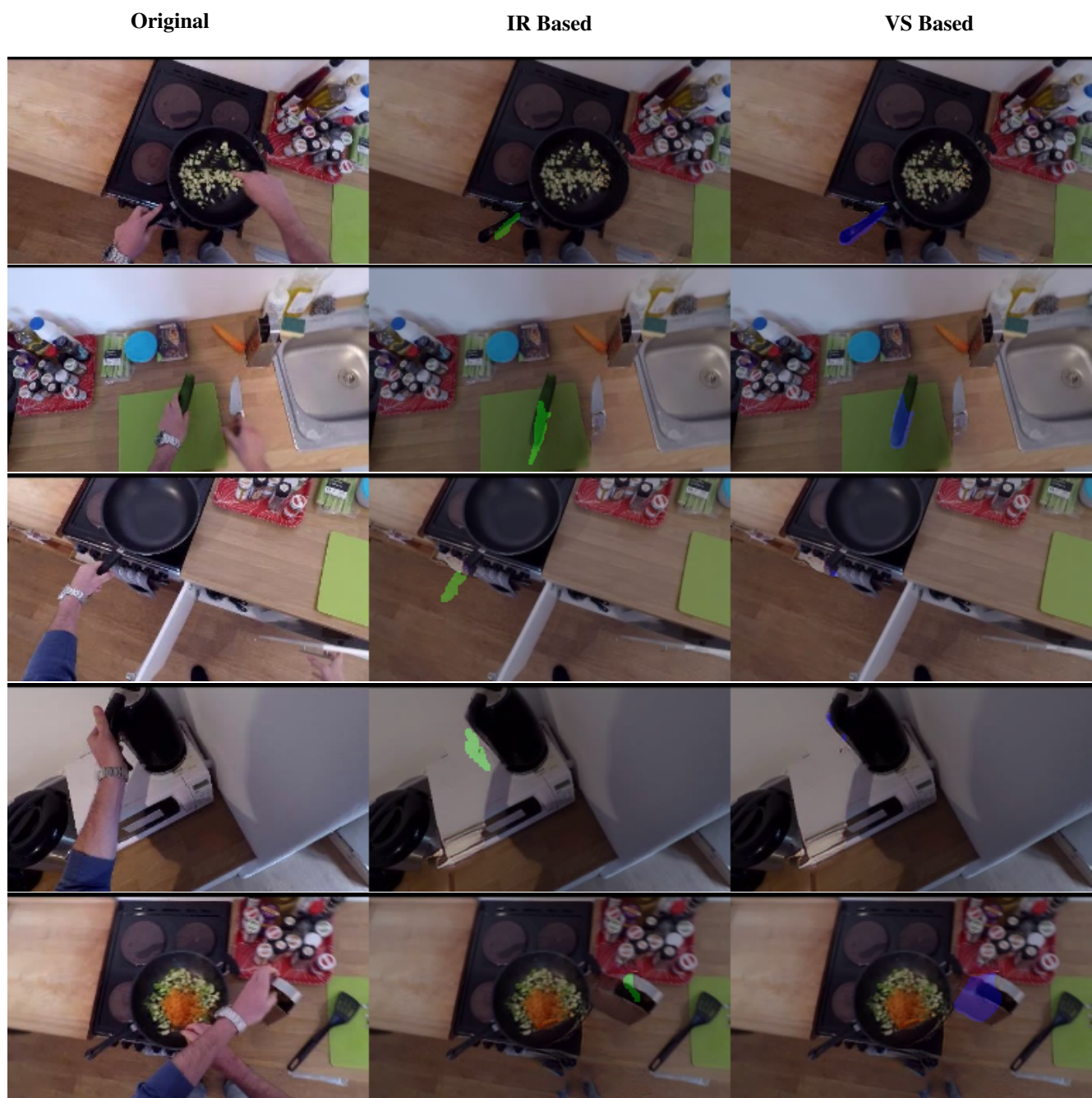


Figure 11. Examples of the two affordance extraction methods. The left column shows the original image, the center column shows the image reconstruction based approach and the right column shows the video segmentation based approach. The video segmentation based approach outperforms the image reconstruction based approach qualitatively in almost every instance.