Pushing it out of the Way: Interactive Visual Navigation –Supplementary Material–

Kuo-Hao Zeng[1] Luca Weihs[2] Ali Farhadi[1] Roozbeh Mottaghi[1,2]
[1]Paul G. Allen School of Computer Science & Engineering, University of Washington
[2]PRIOR @ Allen Institute for AI

A. Heuristic corner detector

Fig. 2 (a) shows our heuristic keypoints detector pipeline. More specifically, we first use an object segmentation model to obtain the segmentation M corresponding to object $o = \texttt{GarbageCan}$. Then, we apply a heuristic corner detector to detect 8 corner points. Note that we use the ground truth segmentation of each object in the training stage, while in the testing stage, we utilize a pretrained MaskRCNN (Sec. C) to obtain the object segmentation. Further we present the details of our heuristic corner detector in Fig. 2 (b), where the 8 corner points are obtained by 8 different criteria and each of the points has to be inside the segmentation M:

$$
\begin{aligned}
p_1 &= \max_{x,\ p_{x,y}\in M} p_{x,y} \\
p_2 &= \max_{y,\ p_{x,y}\in M} p_{x,y} \\
p_3 &= \min_{x,\ p_{x,y}\in M} p_{x,y} \\
p_4 &= \min_{y,\ p_{x,y}\in M} p_{x,y} \\
p_5 &= \max_{x+y,\ p_{x,y}\in M} p_{x,y} \\
p_6 &= \min_{x+y,\ p_{x,y}\in M} p_{x,y} \\
p_7 &= \max_{x-y,\ p_{x,y}\in M} p_{x,y} \\
p_8 &= \min_{x-y,\ p_{x,y}\in M} p_{x,y},
\end{aligned}
$$

where (x, y) is an image coordinate, and M denotes the object segmentation. Based on this heuristic corner detector, we are able to get reliable keypoints from object segmentation. More keypoint examples obtained by our heuristic keypoints detector are shown in Fig. 1.

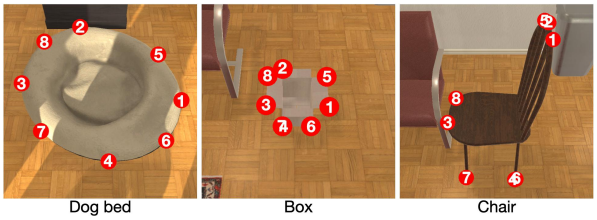

Figure 1: **Keypoints examples.** Examples keypoints obtained by our keypoint detector.

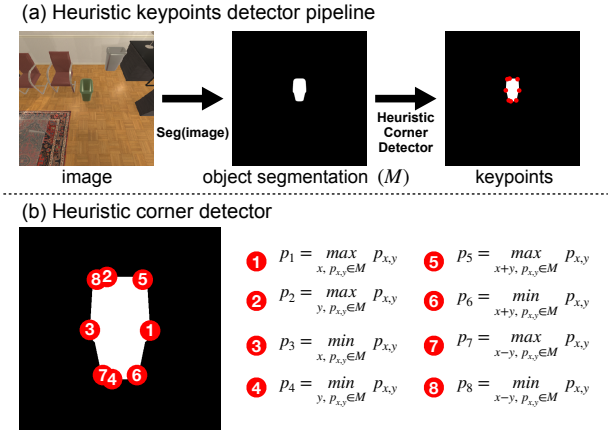

Figure 2: **Keypoint detector details.** (a) Heuristic keypoint detector pipeline. (b) Heurisitc corner detector.

B. Complete list of objects

We use 20 objects for the experiments: *alarm clock*, *apple*, *armchair*, *box*, *bread*, *chair*, *desk*, *dining table*, *dog bed*, *garbage can*, *laptop*, *lettuce*, *microwave*, *pillow*, *pot*, *side table*, *sofa*, *stool*, *television* and *tomato*.

C. MaskRCNN results

We evaluate our pretrained MaskRCNN (ResNet-50 with FPN) on our testing scenes with $\approx$ 2k images. The checkpoint at the 10th epoch achieves 47.4AP and 64.3AP$_{50}$. Fig. 5 shows qualitative results on 20 used objects in one of the testing scenes `LivingRoom227`.

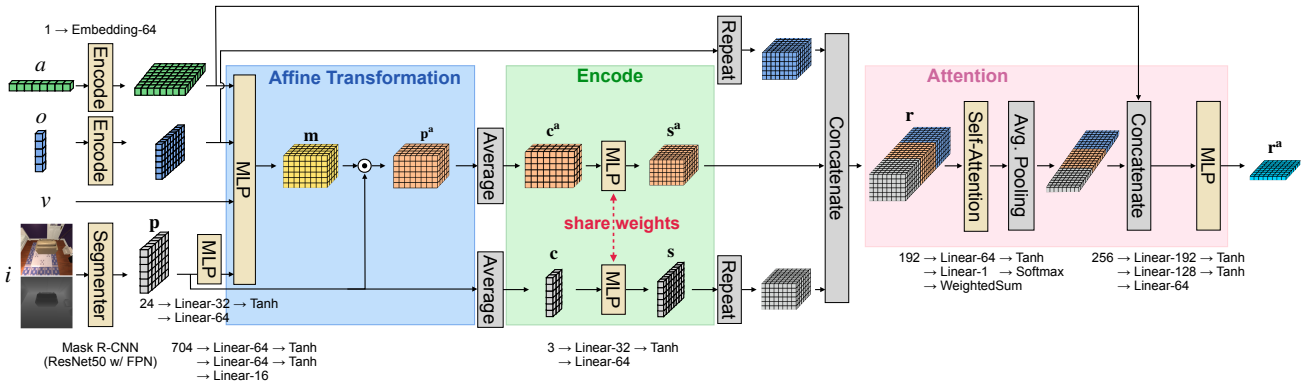

Figure 3: **Detailed architecture of the NIE model.**

D. Details of the model architecture

Fig.3 and Fig. 4 summarize the details of the architecture for visual encoder, goal embedding, policy network, and NIE model.

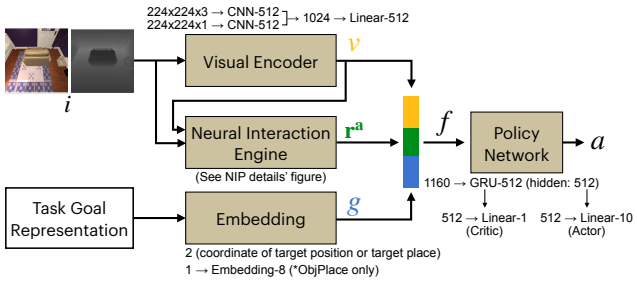

Figure 4: **Detailed architecture of the visual encoder, goal embedding, and policy network.**

E. Action-conditioned keypoints $\mathbf{p}^a$ results

We evaluate our action-conditioned keypoints $\mathbf{p}^a$ prediction on the testing set. Our model achieves 0.148 and 0.114 L1 loss estimation over 8 keypoints on the *ObsNav* and *ObjPlace*, respectively. We found the model performs worse in the *ObsNav* because there are more objects (e.g., obstacles) in this task. Fig. 6 shows the qualitative results of our action-conditioned keypoint prediction.

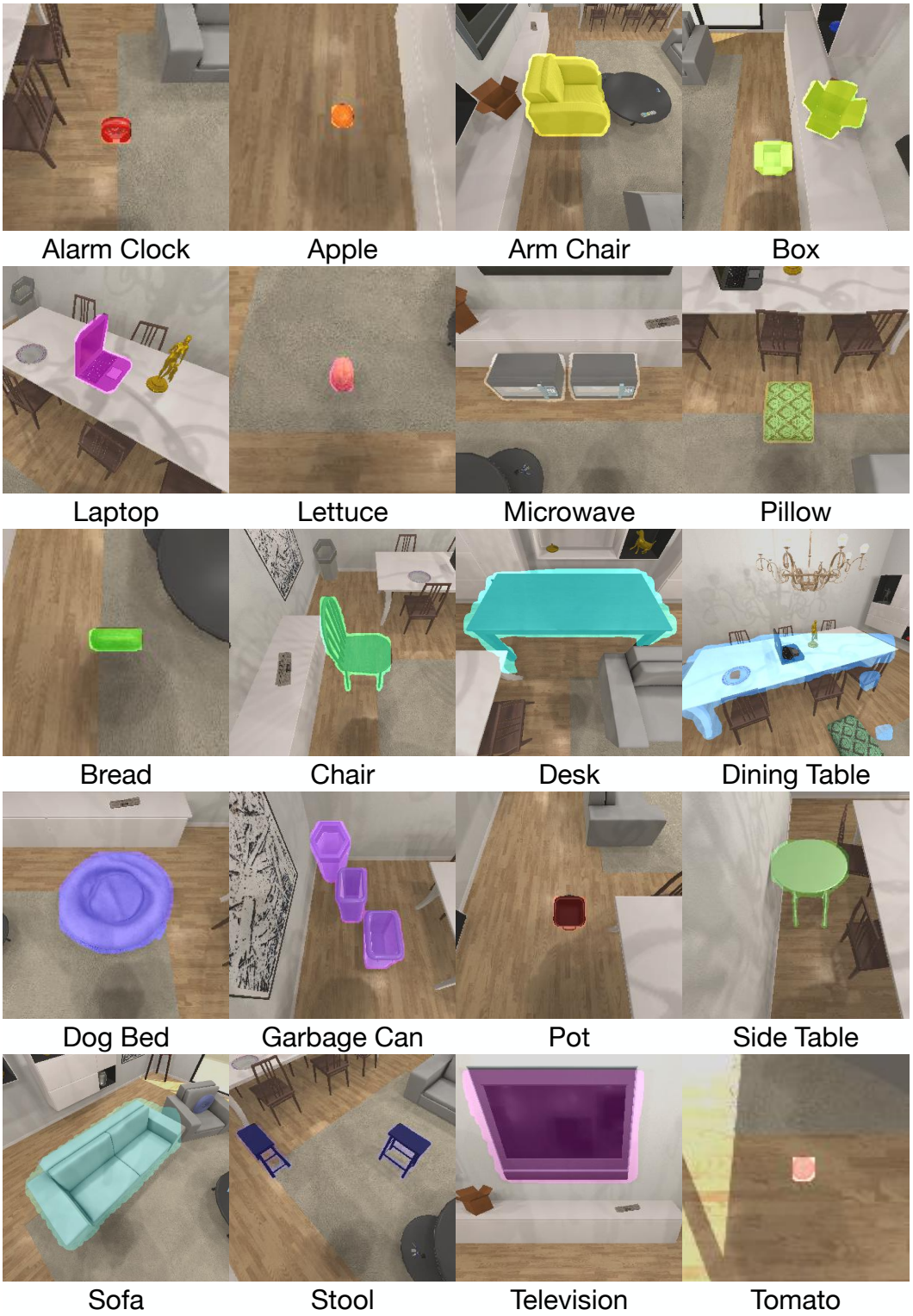

Figure 5: **MaskRCNN's qualitative results on** 20 **used objects.** We randomly spawn 20 objects in the testing scene `LivingRoom227` and apply the pretrained MaskRCNN to obtain the segmentation results. The object prediction score is set to 0.5 and the segmentation probability is set to 0.1.

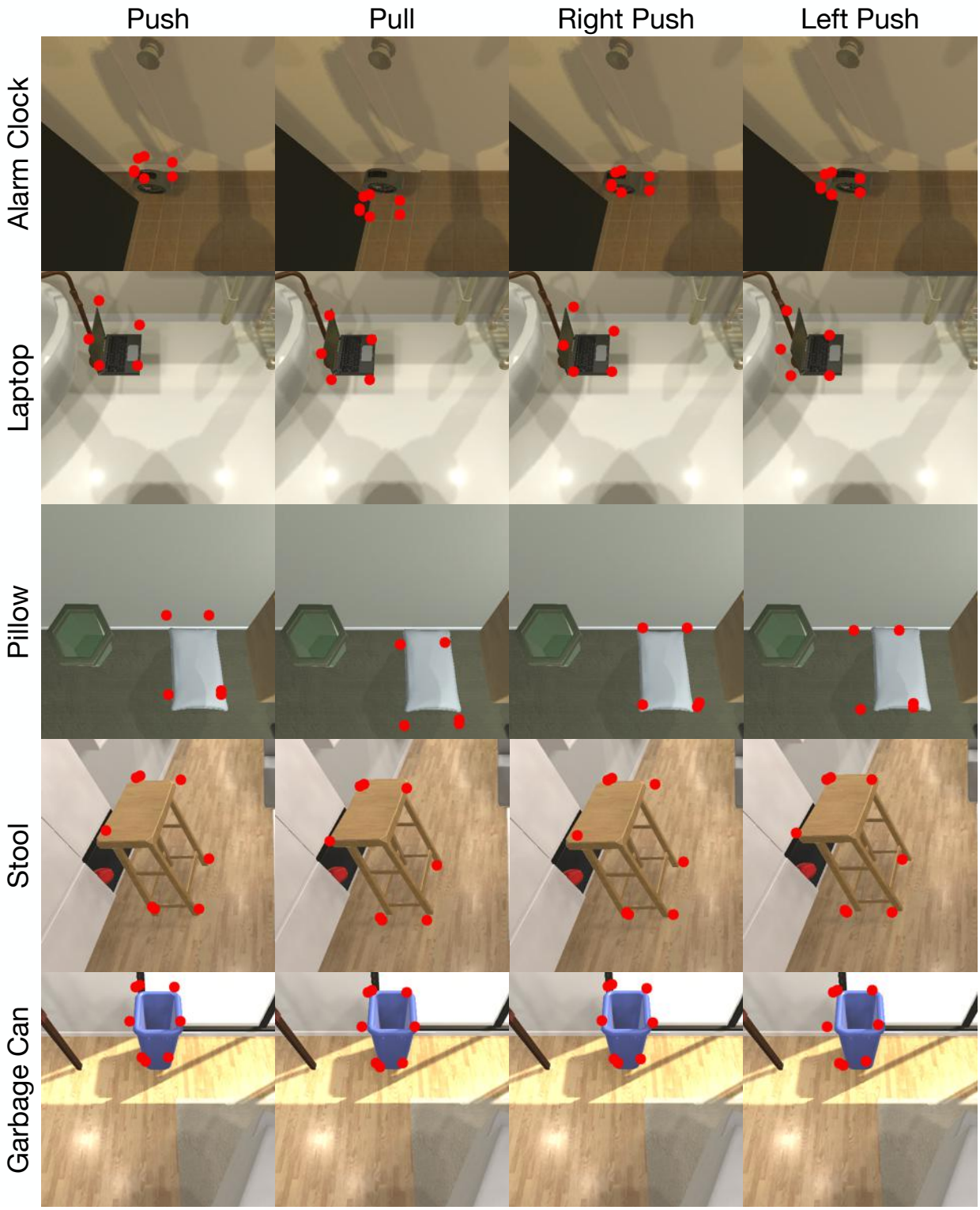

Figure 6: **Qualitative results of action-conditioned keypoints $\mathbf{p}^a$ prediction.** We show our action-conditioned keypoints $\mathbf{p}^a$ prediction results over 4 actions on 5 objects in 4 different testing scene (from top to bottom: `Kitchen27`, `Bathroom430`, `Bedroom328`, and `LivingRoom227`). The predicted keypoints are shown in red color.