## Supplementary Material

We provide supplementary material to complement the main paper. The contents include:

- Section A: We supplement the implementation details of main paper.
- Section B: We present the additional ablation studies for Afformer and MaskAHand.
- Section C: We show MaskAHand generated samples on OPRA [2], EPIC-Hotspot [6], and AssistQ [9] Buttons.
- Section D: We present MaskAHand zero-shot visualization results on OPRA test set.

| Config | Value |
|---|---|
| Optimizer | AdamW, weight decay 0.05 |
| Learning rate | $3 \times 10^{-4}$, cosine decay |
| Training batch size & epochs | 16 batch size in 7 epochs |
| Automatic mixed precision | 16-bit |
| Backbone learning rate factor | 0.1 |
| Backbone Initialization | COCO Detection [5, 8] |
| Attention Heads & Channels | 4 & 256 |
| Maximum Video Frames | 64 by uniform sampling |

(a) Setting for supervised training of Afformer on OPRA [2], EPIC-Hotspots [6], and AssistQ [9] Buttons. Note that EPIC-Hotspots and AssistQ Buttons are only experimented with R50-FPN backbone.

| Config | Value |
|---|---|
| Backbone | R50-FPN [3, 4] |
| Batch size | 32 |
| Mined video frames | 32 by sequential sampling |
| Stride between mined videos | 16 frames |
| Interaction score threshold | 0.99 |

(b) Setting for MaskAHand pre-training on OPRA [2], EPIC-Hotspots [6], and AssistQ [9] Buttons. Other settings follow (a).

Table 1. Implementation details of Afformer and MaskAHand.

| Module | I,V Shared | KLD ↓ |
|---|---|---|
| Backbone | × | 1.80 |
| | ✓ | **1.57** |
| Input Proj | × | 1.65 |
| | ✓ | **1.57** |

(a) Sharing parameters for image and video encoding has better results.

| Module | Pyramid Shared | KLD ↓ |
|---|---|---|
| Input Proj | × | **1.57** |
| | ✓ | 1.60 |
| Decoder | × | 1.61 |
| | ✓ | **1.57** |

(b) Different pyramids need different input projections, but the decoder can be shared.

Table 2. Additional ablation studies for Afformer on OPRA.

| Mask | Zero-shot KLD ↓ | Fine-tune KLD ↓ |
|---|---|---|
| Zero (0) | 2.65 | 1.50 |
| Random ($0 \sim 255$) | **2.36** | **1.48** |

Table 3. Experiments on masked fill value of MaskAHand pre-training on OPRA.

## A. Implementation Details

Due to limited pages, some implementation details are not presented in the main paper. We complement them in Table 1. The initialization for Afformer's backbone follows Demo2Vec [2], which also initializes the backbone network by COCO detection [5] weights. We use `PyTorch` [7] and `torchvision` to implement our model.

## B. Additional Ablation Studies

Table 2 shows the additional ablation studies for Afformer, which suggests that the backbone and input projector should be shared for video and target image. Furthermore, we observe that the input projector should be different for different pyramid resolution levels, but the transformer decoder can be shared. The sharing of the backbone and decoder makes our Afformer parameter efficient.

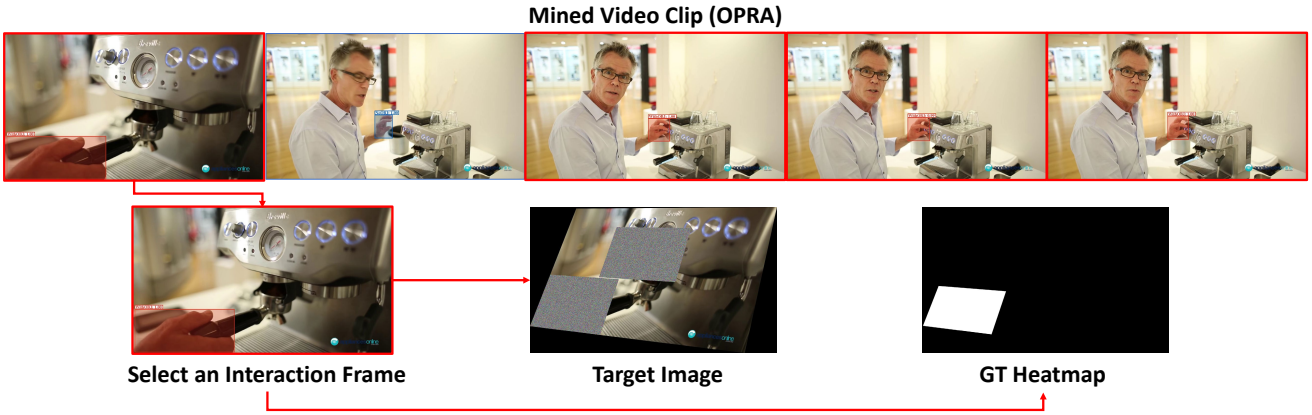
We also investigate the masked filled value of MaskAHand pre-training, as shown in Table 3. In the masked region of the target image, filling random value (*i.e.* random noise mask) is much better than zero value (*i.e.* black mask). We hypothesize that the fixed zero filling makes model overfitting easier.

## C. Training Samples from MaskAHand

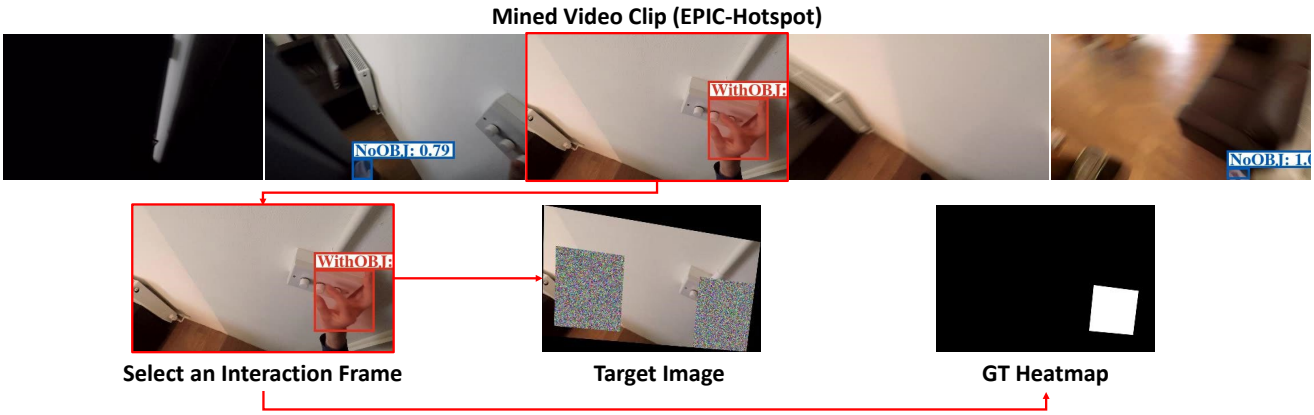
Figure 1 shows MaskAHand generated training samples on different datasets. Each training sample include a video clip, a target image, and a ground-truth heatmap.

## D. MaskAHand Zero-shot Visualization

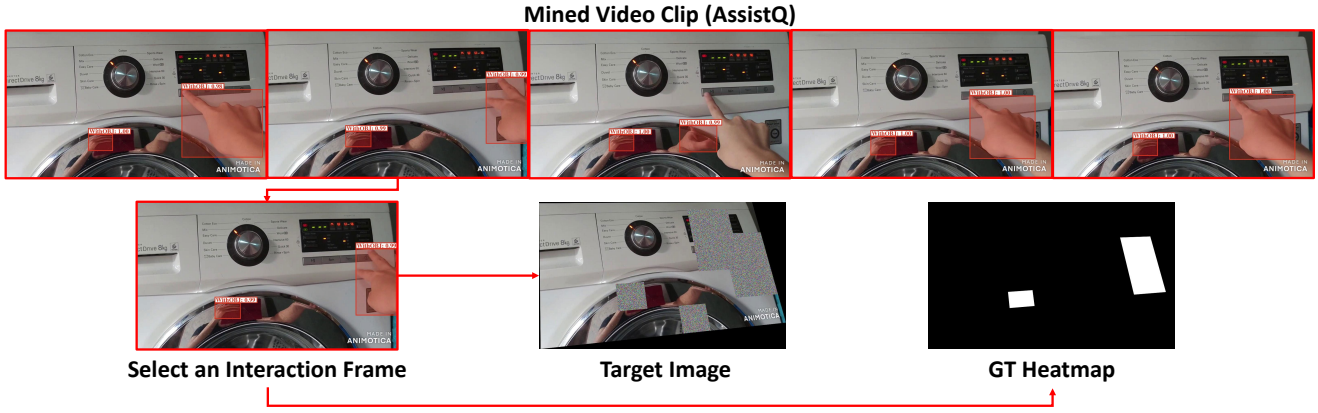
Figure 2 visualizes MaskAHand zero-shot results, demonstrating that the representation learned by MaskAHand can support video-to-image affordance grounding.




(a) A training sample case generated by MaskAHand, on OPRA training set's raw videos.




(b) A training sample case generated by MaskAHand, on EPIC-Hotspots training set's raw videos (*i.e.* EPIC-KITCHEN frames [1]).




(c) A training sample case generated by MaskAHand, on AssistQ training set's raw videos.

Figure 1. MaskAHand generated training sample cases on OPRA [2], EPIC-Hotspots [6], and AssistQ [9]. The hand interaction detection is made by our trained hand interaction R-CNN, mentioned in Section 4.2. "WithObj" means the hand is interacting with objects, whereas "NoObj" means not interacting. The mined video clip contains 32 frames (with 5 FPS). If there are multiple interaction frames detected (*e.g.*, (a) and (c)), the target image generation will be randomly picked from these frames.

## References

[1] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The dataset. In *ECCV*, pages 753–771, 2018. 2

[2] Kuan Fang, Te-Lin Wu, Daniel Yang, Silvio Savarese, and Joseph J. Lim. Demo2vec: Reasoning object affordances from

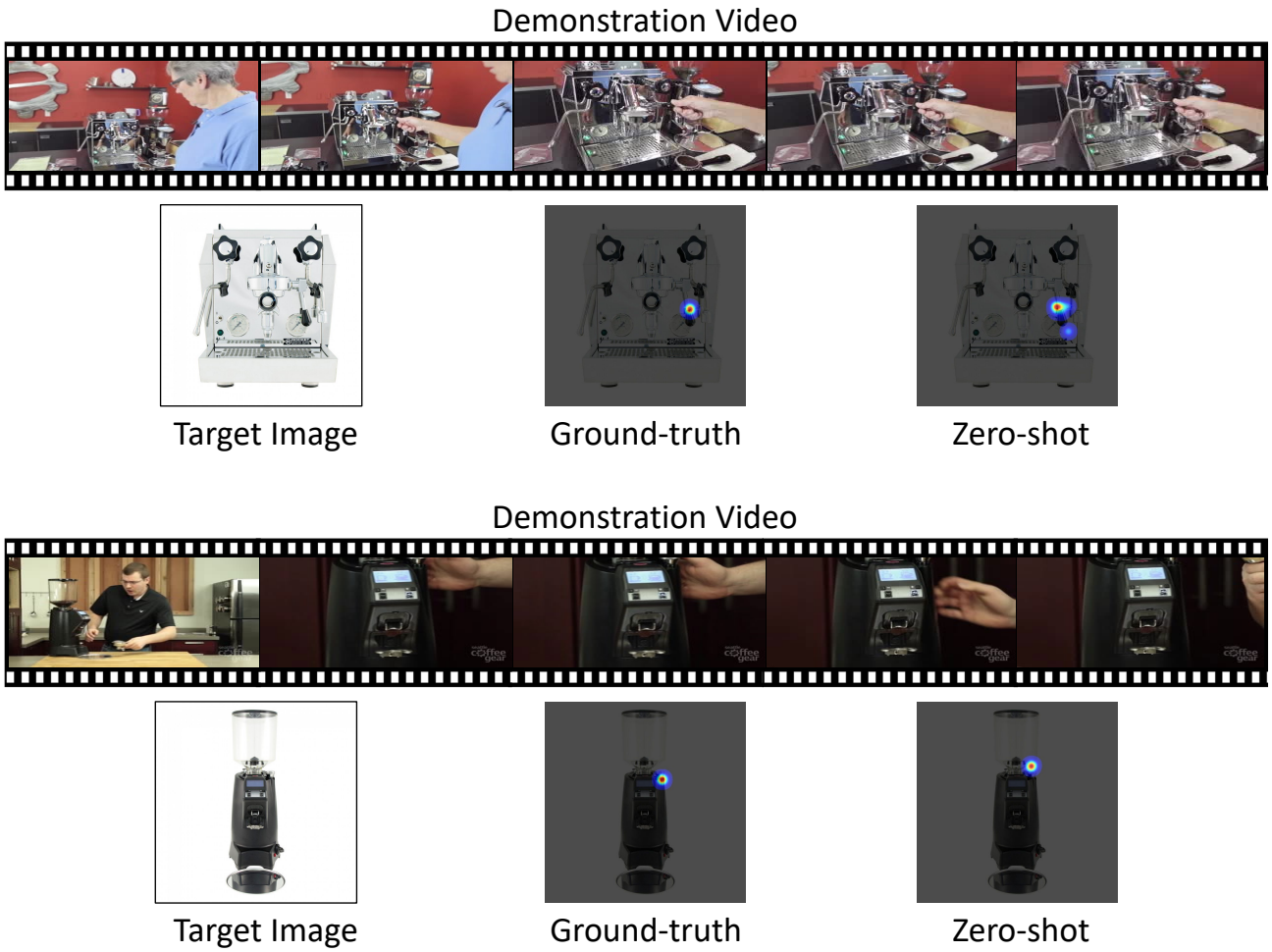



Figure 2. MaskAHand zero-shot results on OPRA test set. For visualization, we select the top 100 points on the ground-truth and zero-shot prediction heatmap. It can be seen that the hotspot on the zero-shot heatmap is close to that on the ground-truth heatmap.

online videos. In *CVPR*, pages 2139–2147, 2018. 1, 2

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[4] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 936–944, 2017. 1

[5] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, pages 740–755, 2014. 1

[6] Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. Grounded human-object interaction hotspots from video. In *ICCV*, pages 8687–8696, 2019. 1, 2

[7] Adam Paszke, Sam Gross, and Francisco et al Massa. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8026–8037, 2019. 1

[8] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017. 1

[9] Benita Wong, Joya Chen, You Wu, Stan Weixian Lei, Dongxing Mao, Difei Gao, and Mike Zheng Shou. Assistq: Affordance-centric question-driven task completion for egocentric assistant. In *ECCV*, pages 485–501, 2022. 1, 2