

## A SUPPLEMENTARY MATERIAL



Figure A1: **VISOR-HOS performance on non-inpainted vs. inpainted FPHAB.** We visualize VISOR-HOS’ output on the non-inpainted RGB image (left) and the corresponding inpainted image (right).

### A.1 GENERATION OF VISUALIZATIONS.

The hand-object trajectory visualizations in this work are generated by first performing an iterative optimization minimizing the distances between predefined hand mesh vertices and object contact points, then updating the rotation of the hand to follow the rotation of predefined finger joint pairs (e.g. the wrist-to-index rotation in case of “pour milk”).

### A.2 INPAINTING FPHAB MOTION CAPTURE MARKERS.

The performance of the employed hand-object interaction detection models suffers significantly from the motion capture (mocap) markers visible on the subjects’ hands in FPHAB. To counteract the problem, we train a Lawin (Yan et al. (2022)) model to segment the mocap markers in each frame after manually creating a small segmentation dataset, then use the E2FGVI (Li et al. (2022)) model to inpaint the mocap markers away after segmenting them on the entire dataset. Visualizations are provided in Figure A1.

### A.3 BASELINES

We evaluate against the following baselines in Table 1:

- **T2M-T** (Guo et al. (2022)): Similar to the original work, we first train a motion autoencoder to work with motion snippets. Next, we train a text encoder along with the decoder part of the motion autoencoder. We then integrate these components into a triplet variational autoencoder network to synthesize motions. As the original method is heavily adapted to textual conditioning, we only perform a text-based evaluation.
- **MDM-T** (Tevet et al. (2022)): MDM (Motion Diffusion Model) uses a Transformer embedded into a denoising process that gradually refines random noise into motions based on conditioning information, as described in Sec. 3.3. For MDM-T, we consider the original, text-based framework.
- **MDM-I** (Tevet et al. (2022)): This setup also uses MDM, but conditions on image features extracted from the *entire* image, without cropping to any object.

### A.4 HYPERPARAMETERS

**MDM.** Our MDM (Tevet et al. (2022)) models are trained for 250K steps, using a batch size of 32 and a learning rate of  $10^{-4}$  with the Adam optimizer. We set MDM’s  $T$  parameter to  $T = 1000$  to perform 1000 diffusion steps per motion, and further use  $\lambda_{pos} = \lambda_{vel} = 1, \lambda_{foot} = 0$  for MDM. When evaluating and generating motions, we set the number of frames to generate to  $n_f = 200$  so as to cover the vast majority of the motion length distribution of both datasets. A guidance parameter of  $s = 2.5$  and a latent embedding size of  $d = 512$  is used.

action	original task(s)	object categories	instances in training set	instances in test set
bind	bind the paper	stapler	27	11
clamp	clamp something	pliers	23	9
cut	cut something	scissors	28	12
enable/disable	turn on and turn off	lamp	-	-
fill	fill with water by a kettle	mug	7	2
move	pick and place	bottle, bowl, bucket, kettle, knife, lamp, laptop, mug, pliers, scissors, stapler, toy car	18, -, 23, 28, 28, -, 25, 25, 28, 26, 28, -	5, -, 9, 12, 12, -, 9, 11, 11, 10, 12, -
	pick and place to original position	chair	-	-
	pick and place to new position	chair	-	-
	pick and place with ball	bowl	-	-
	pick and place with water	bottle, mug	16, 13	8, 3
open/close	open and close	trash can	-	-
	open and close the display	laptop	33	13
	open and close the door	storage furniture	-	-
	open and close the drawer	storage furniture	-	-
pour	pour all the water into a mug	bottle	24	9
	pour water into a mug	kettle	28	12
	pour water into another mug	mug	14	5
	pour water into another bucket	bucket	27	11
push	push toy car	toy car	-	-
retrieve	take it out of the drawer	bottle, bowl, knife, pliers, toy car	19, -, 12, 23, -	8, -, 4, 9, -
	take something out of it	safe	-	-
	take the ball out of the bowl	bowl	-	-
slice	cut apple	knife	26	10
store	put it in the drawer	bottle, bowl, knife, mug, pliers	10, -, 15, 19, 19	4, -, 6, 8, 7
	put the drink in the door	storage furniture	-	-
	put the drink in the drawer	storage furniture	-	-
	throw something in it	trash can	-	-
turn	turn and fold	lamp	-	-

Table A1: **Actions for the HOI4D instance split.** Our proposed action-centric *instance split* for the HOI4D dataset, assigning each instance to either the train or the test set for each object and action. In this work, we train and evaluate on the objects *bottle*, *bucket*, *kettle*, *knife*, *laptop*, *mug*, *pliers*, *scissors*, *stapler*, retaining a total of 10 from the 13 proposed actions.

---

**Part feature extraction.** We use the `scipy.ndimage.binary_dilation` function of the `scikit-learn` package to compute the dilated hand masks. 7 dilation iterations are used to obtain the first mask, and 15 dilation iterations are used to obtain the second mask. The second mask hence fully includes and extends the first mask. A single dilation iteration can be thought of as approximately isotropically extending the mask obtained by the previous dilation operation. Please see Dougherty (1992) for the full description of a dilation iteration.

We use 2D CLIP feature grid patches falling into the second dilated hand mask, but not the first dilated hand mask, to compute the part feature vectors. At the same time, the used patches must also fall into the region of the object mask, which we do not interpolate. The idea is to include rectangular patches close to the contact region (use of the dilated hand mask) and limited to the object (use of the object mask), but not showing the hand/fingers (subtraction of the undilated hand mask from the object mask).

Whenever part features are employed, they are used in combination with object features. This is accomplished by providing a separate part feature conditioning token together with the always-provided object feature conditioning token to the motion generator.

## A.5 HOI4D INSTANCE SPLIT

We list the *actions* derived from the original tasks proposed by HOI4D, as well as the corresponding instance counts per action and object category in the training and test set of our HOI4D instance set in Table A1.

## A.6 DETAILS ON SIMULATION EXPERIMENTS

### A.6.1 HAND TRAJECTORY RETARGETING

We use the Adroit hand model employed by Rajeswaran et al. (2017) to evaluate our trajectories in the simulator. This necessitates a conversion of the SMPL-X hand trajectory to a functionally equivalent trajectory where the Adroit hand is used. One of the commonly employed pagadigms for this conversion is introduced in Shaw et al. (2022). It proposes solving a quadratic optimization problem during each timestep to solve for the joint rotations of the robotic hand. The optimization aims to make the distances between the human fingertips observed during the manipulation match the distances between the agent’s fingertips, as well as align the palm-fingertip distances in the same manner. Temporal smoothing is further used to stabilize the resulting retargeted trajectory. Please consult Shaw et al. (2022) for further details.

### A.6.2 HAND TRAJECTORIES TO ACTUATOR ACTIVATIONS

The MuJoCo simulator requires the simulation script to provide actuator activations during each environment step. This necessitates a conversion from the 6D hand joint representation our model employs to a representation of the motion in terms of actuator activations.

### A.6.3 GRASP INITIALIZATION

As the synthesis of a grasp that holds the object in place and avoids hand-object penetration is highly challenging and outside the scope of our work, we initialize the hand trajectories evaluated in the simulator with manually defined grasps. After the initialization, we follow the trajectory synthesized by the respective model by applying the calculated joint actions, as described in subsection A.6.2. Additionally, we apply a strong bias on multiple mid-finger actuators, which causes the hand to squeeze the held object and thereby prevent the object from slipping out of the hand.

## A.7 TRAINING OF VISOR-HOS

The VISOR-HOS (Darkhalil et al. (2022)) we use to identify interacted objects is trained on a subset of the egocentric EPIC-KITCHENS 100 (Damen et al. (2022)) dataset. The subset is a combination of a small number of frames with manually annotated interacted object masks, and a large number of frames with interacted object masks automatically interpolated using the sparse manually annotated masks. The class of the interacted objects is not predicted. In total, 14.5M masks are used

---

during training and 3.2M are used during validation to train the segmentation model. Please refer to Darkhalil et al. (2022) for further details.

## A.8 DATASET SIZES

After processing the videos, we obtain 3860 resp. 542 samples for the train resp. validation split of the FPHAB dataset. For the HOI4D dataset, we obtain 3954 resp. 169 samples for the instance split, and 5014 resp. 104 samples for the site split.

## A.9 COMPUTE RESOURCES

For training the MDM-based diffusion models, we used an RTX-4090, and trained for 36 hours and 250,000 epochs. For evaluation inside the simulator, we used a AMD EPYC 7742 CPU with 128 cores. Evaluating 50 trajectories takes around 2 hours, with the *pour milk* task taking the longest.

## REFERENCES

- Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling Egocentric Vision: Collection, Pipeline and Challenges for EPIC-KITCHENS-100. 130:33–55, 2022. URL <https://doi.org/10.1007/s11263-021-01531-2>.
- Ahmad Darkhalil, Dandan Shan, Bin Zhu, Jian Ma, Amlan Kar, Richard Higgins, Sanja Fidler, David Fouhey, and Dima Damen. EPIC-KITCHENS VISOR Benchmark: Video Segmentations and Object Relations. In *Proceedings of the Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*, 2022.
- Edward R Dougherty. An Introduction to Morphological Image Processing. In *SPIE. Optical Engineering Press*, 1992.
- Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating Diverse and Natural 3D Human Motions From Text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5152–5161, June 2022.
- Zhen Li, Cheng-Ze Lu, Jianhua Qin, Chun-Le Guo, and Ming-Ming Cheng. Towards an End-To-End Framework for Flow-Guided Video Inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17562–17571, 2022.
- Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning Complex Dexterous Manipulation With Deep Reinforcement Learning and Demonstrations. 2017.
- Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. VideoDex: Learning Dexterity From Internet Videos, 2022.
- Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human Motion Diffusion Model. 2022.
- Haotian Yan, Chuang Zhang, and Ming Wu. Lawin Transformer: Improving Semantic Segmentation Transformer With Multi-Scale Representations via Large Window Attention. 2022.