

# VidBot: Learning Generalizable 3D Actions from In-the-Wild 2D Human Videos for Zero-Shot Robotic Manipulation

## Supplementary Material

### A. Supplementary Video

We include a supplementary video showcasing an overview of our framework, *VidBot*, along with demonstrations of various real-world robot manipulation tasks.

### B. Diffusion Models Formulation

#### B.1. Denoising Prediction

As our denoising network learns a conditional distribution, hence Eqn. 4 will be re-written as:

$$\begin{aligned} p_\phi(\tau^{k-1}|\tau^k) &= \mathcal{N}(\tau^{k-1}; \mu_\phi(\tau^k, k, o), \Sigma_k), \\ p_\phi(\tau^{1:K}) &= p(\tau^K) \prod_{k=1}^K p_\phi(\tau^{k-1}|\tau^k), \end{aligned} \quad (12)$$

where  $\mu_\phi(\tau^k, k, o)$  is obtained through the denoising neural network, we denote it as  $\mu^k$  for simplicity, and  $\Sigma_k$  is from a fixed scheduler,  $o$  is the task observations we introduced in the main paper.

Following Eqn. 12, we need to acquire  $\mu^k$  so that  $\tau^{k-1}$  could be sampled from  $\mathcal{N}(\tau^{k-1}; \mu_\phi(\tau^k, k, o), \Sigma_k)$ . As our denoising network directly predicts the unnoised trajectory  $\bar{\tau}^0$ , so we follow the scheme shown in [67] to acquire  $\mu^k$ :

$$\mu^k = \frac{\sqrt{\alpha_k} \beta_k}{1 - \alpha_k} \bar{\tau}^0 + \frac{\sqrt{\alpha_k} (1 - \alpha_k)}{1 - \alpha_k} \tau^k, \quad (13)$$

where  $\beta_k$  is the variance following a cosine schedule [67],  $\alpha_k = 1 - \beta_k$ ,  $\bar{\alpha}_k = \prod_{j=0}^k \alpha_j$ .

Moreover, we can retrieve the added noise  $\epsilon^k$  in each step  $k$  as:

$$\begin{aligned} \tau^k &= \sqrt{\alpha_k} \bar{\tau}^0 + \sqrt{1 - \alpha_k} \epsilon^k, \\ \epsilon^k &= \frac{\tau^k - \sqrt{\alpha_k} \bar{\tau}^0}{\sqrt{1 - \alpha_k}}, \end{aligned} \quad (14)$$

where  $\epsilon^k \sim \mathcal{N}(0, \mathbf{I})$ .

We can leverage  $\epsilon^k$  to conduct a classifier-free diffusion guidance strategy [36] to acquire the trajectory outputs during test time.

#### B.2. Classifier-free Diffusion

With a slight abuse of notation, we rewrite  $\epsilon^k$  as  $\epsilon_\phi(\tau^k, k, o)$ , as we have shown in Eqn. 13 and Eqn. 14 that  $\epsilon^k$  can be recovered from denoising network's output differently. Hence,  $\epsilon^k$  can also be treated as the conditional prediction. We also acquire the unconditional prediction from

the network by dropping the  $o$ , i.e.,  $\epsilon_\phi(\tau^k, k)$ . Hence, following the classifier-free diffusion guidance strategy from [36], we acquire the final predictions of the  $\tilde{\epsilon}^k$ :

$$\tilde{\epsilon}^k = \epsilon_\phi(\tau^k, k, o) + w(\epsilon_\phi(\tau^k, k, o) - \epsilon_\phi(\tau^k, k)) \quad (15)$$

$w$  is used to balance the strength between conditional and unconditional sampling. This strategy has been shown to improve the diversity of the sampled results and capture the underlying distribution of the training data [80], benefiting the cost-guided trajectory generation during the test time.

### C. Implementation Details

#### C.1. 3D Affordance Data Extraction Details

Here, we showcase the implementation details of the 3D affordance data extraction pipeline. We leverage the EpicKitchens-100 Videos dataset [20] to showcase the effectiveness of our pipeline. This dataset comprises hundreds of hours of video recording in which humans perform everyday household skills in diverse kitchen environments. This dataset is in an *in-the-wild* setting, as no ground-truth 3D information like depth or pose is accessible. It's particularly worth noting that this dataset is not collected for robot learning tasks.

To obtain each 3D affordance label sample, we work with video clips with narration like *Open the cupboard* or *Wipe the counter* as shown in Fig. 2, which provide us with an image sequence  $\{\hat{\mathbf{I}}_1, \dots, \hat{\mathbf{I}}_T\}$  and language description  $l$ . We use the pre-computed SfM results from [89] to obtain camera intrinsics  $\mathbf{K}$ , per-frame's pose  $\{\mathbf{T}_{\text{WC}_1}, \dots, \mathbf{T}_{\text{WC}_T}\}$  and sparse landmarks  $\{\mathbf{w}_1, \dots, \mathbf{w}_{N_l}\}$  expressed in the world frame. The dataset also provides 2D bounding boxes of the hands and the in-contact objects acquired using [84]. We leverage [108] to acquire the full hands' masks and use the object's bounding box to prompt SAM [48] to acquire objects' masks, i.e.,  $\{\mathbf{M}_1^h, \dots, \mathbf{M}_T^h\}$ ,  $\{\mathbf{M}_1^o, \dots, \mathbf{M}_T^o\}$ .

Though the language description is manually provided along with the dataset, LLM like [79] has shown to be highly effective in automating the video clip retrieval and narration generation process, as demonstrated in [71]. Since automating the language description generation process is not the focus and contribution of our work, we leave it for future work.

**Trajectory extraction.** To optimize the Eqn. 1, we first collect the scale offset value of each frame computed by comparing the median depth of all valid projected landmarks in the camera frame and the median value of their

corresponding predicted metric depth. We then choose the median value from the collected scale offset values to initialize  $s_g$ . We leverage an Adam optimizer with a learning rate of 0.05 and optimize this loss term for 10 iterations.

For the Eqn. 2, we initialize each frame’s pose using the pre-computed SfM results and its scale using the optimized global scale value  $s_g$ . We leverage an Adam optimizer with a learning rate of 0.1 for scale and a learning rate of 0.05 for pose. We use the continuous representation [111] to parameterize rotation. They are optimized for 50 iterations.

After optimization, we obtain each frame’s hand center point and transform it to the first frame with the optimized poses and scales to compute the interaction trajectory  $\hat{\tau}$ . As these trajectory waypoints are discrete, we fit a spline curve to these points and then sample 80 points uniformly along the curve to obtain the final dense trajectory. We also adopt the Savitzky–Golay filters [82] to further smooth the trajectory. We finally obtained around 50k training samples. Fig. 11 demonstrates more examples of our extracted training data.

**Contact points extraction.** We uniformly downsample points from the in-contact hand in the first frame to acquire contact points  $\hat{c}$  and project them to the image plane, then fit a GMM to the pixel coordinates with 4 clusters. We then center a Gaussian distribution over each cluster center to acquire the contact heatmaps. We include an auxiliary vector field label to supervise the contact predictor. The vector field labels are computed using per-pixel directions pointing to each cluster center, which yields 4 vector field labels for each training sample. Fig. 12 demonstrates the contact heatmap and the vector fields pointing to each cluster center.

**Goal points extraction.** Similar to contact points extraction, we project the goal points uniformly downsampled from the hand to the first frame and center one Gaussian distribution to the points. Note that we only have one cluster center; hence, we only need to compute one vector field label. Fig. 13 demonstrates the goal heatmap and the associated vector field.

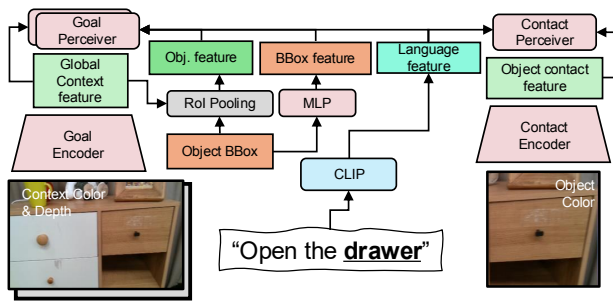


Figure 7. Illustration of conditional feature extraction for the goal predictor and the contact predictor.

## C.2. Network Architecture

Here, we detail the architectural design for each model.

**Goal predictor architecture.** Our goal predictor follows a similar UNet architecture as [37]. It takes a 4-channel input with a resolution of  $256 \times 448$ ; the first 3 channels are RGB values, and the last is filled with the median depth value of the object of interest. The goal encoder has 5 convolutional residual blocks that down-sample the resolution by a factor of 2 between layers, and self-attention blocks are applied between the convolutional blocks to the global context feature  $z^{\text{goal}}$  with size being the  $1/16$  of the original resolution, yielding dimension of 512. The bounding box positional feature  $z_b^{\text{goal}}$  has a dimension of 128 after being passed to an MLP layer. The language feature  $z_l$  has a dimension of 512 after being passed to the frozen CLIP model [76] and an MLP layer. We apply RoI Pooling on the global context feature  $z^{\text{goal}}$ , followed by average pooling, to obtain the object feature  $z_o^{\text{goal}}$ , yielding a dimension of 512 as well; we further pass it to an MLP layer as well, without changing its dimension. We obtain the conditional feature,  $z_c^{\text{goal}} = \{z_o^{\text{goal}}, z_b^{\text{goal}}, z_l\}$  with dimension of 1152. We then fuse the conditional feature  $z_c^{\text{goal}}$  to the visual context feature  $z^{\text{goal}}$  using a Perceiver module [40]. Fig. 7 illustrates how we obtain the conditional feature.

Finally, a symmetric decoder is applied to up-sample the fused context feature to the inputs’ resolution, which gives 3 channel outputs, with the first channel being the predicted goal probabilities and the rest being the auxiliary vector field values. To predict the depth value of the goal points, we extract the language feature of the verbs of the language instructions (e.g., “open,” “pick-up”) and apply another Perceiver module [40] to fuse it to the global context feature. We further pass the fused feature to a block with 3 transformer encoder layers possessing 4 heads and a feedforward dimension of 512, followed by an MLP layer to acquire the goal depth value.

**Contact predictor architecture.** The contact predictor takes an object color image with a resolution of  $256 \times 256$  as inputs and outputs 9-channel predictions, with the first channel being the predicted contact probabilities and the rest being the 4 vector fields as auxiliary predictions (each vector field has two channels). We leverage a ResNet50 [35] encoder as our contact encoder, which provides features with dimensions of 64, 256, 512, 1024, and 2048. The bottleneck feature has a dimension of 2048. We first pass it to an MLP layer to acquire visual features with a dimension of 512; we also pass the language feature to an MLP so that it has a dimension of 512. We then fuse the language feature to the visual feature using a Perceiver module [40]. Fig. 7 illustrates how we obtain the conditional feature.

After the feature fusion, we project the fused latent feature back to the dimension of 2048 with a block comprising one transformer encoder layer with 4 heads and a feedfor-

ward dimension of 512 and an MLP layer. We repeatedly perform skip connection, convolution, and up-sampling on the feature map during decoding process so that the outputted result reaches the input color’s resolution.

**Fine affordance predictor architecture.** As shown in Fig. 3, we apply sinusoidal positional encoding to the denoising step  $k$ , goal point  $\mathbf{g}$  with the highest probability and the contact point  $\bar{\mathbf{c}}$  with the highest probability, yielding feature with dimensions of 32, 48, 48. The color feature of the object of interest,  $\mathbf{z}_o^{\text{fine}}$ , extracted using ViT-based feature extractor [10], has a dimension of 256. The language feature is also 256-dimensional, achieved through a projection layer. This projection layer consists of a transformer encoder layer with 4 heads and a feedforward dimension of 512, together with a final MLP layer that projects the 512-dimensional language feature down to 256 dimensions. The 3D-UNet TSDF feature extractor consists of three consecutive blocks comprising 3D convolution, ReLU [66], and GroupNorm [97] layers to produce the TSDF feature grids with dimensions of 64. Hence, this yields the conditional feature  $\mathbf{o} = \{\text{PE}(\mathbf{g}), \text{PE}(\bar{\mathbf{c}}), \text{Proj}(\mathbf{z}_l), \mathbf{z}_o^{\text{fine}}\}$  with a dimension of 608; the denoising inputs  $\mathbf{x}^k = \{\boldsymbol{\tau}^k, \mathbf{f}^k\}$  fed to  $\pi_f$  with a dimension of  $H \times 67$ ,  $H = 80$  being the trajectory horizon. The encoded denoising step feature  $\text{PE}(k)$  has a feature dimension of 32.

In the 1D U-Net Denoiser, we first concatenate the conditional feature  $\mathbf{o}$  with the denoising step feature  $\text{PE}(k)$  to form one conditional feature, which we then fuse with the denoising input  $\mathbf{x}^k$  (the trajectory state) using a Perceiver module [40]. Given the conditional feature  $\mathbf{o}$ , the step feature  $\text{PE}(k)$ , and the denoising inputs  $\mathbf{x}^k$  after Perceiver fusion, we adopt a 1D UNet similar to [80] with temporal convolutions over the trajectory horizon dimension. The temporal dimension is down-sampled by a factor of 2 between layers. Conversely, in the decoding section, a  $2\times$  up-sampling is applied. The denoiser encoder has four convolutional layers with output dimensions of 32, 64, 128, and 256, where each layer has 2 residual blocks. Fig. 8 illustrates the architecture of one single residual block.

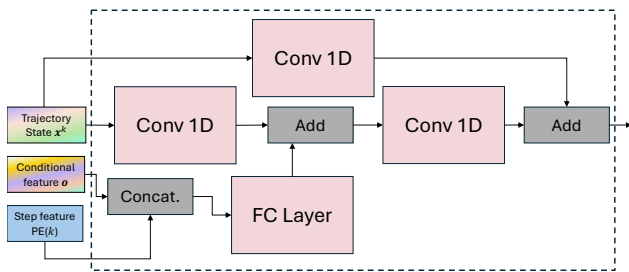


Figure 8. Architecture illustration of one residual block in the denoiser network.

### C.3. Training Details

Here, we detail the training protocols for each model of our affordance prediction pipeline.

**Goal predictor training.** We train the goal predictor network for 30k iterations using Adam optimizer [47] with a learning rate  $1e-4$ , and batch size 12. We set  $\lambda_d = 1$ ,  $\lambda_v = 0.1$ .

**Contact predictor training.** We train the contact predictor network for 30k iterations using Adam optimizer [47] with a learning rate  $5e-5$ , and batch size 8. We set  $\lambda_v = 0.1$ .

**Fine affordance predictor training.** We train the fine affordance predictor network for 30k iterations using Adam optimizer [47] with a learning rate  $1e-4$ , and batch size 8. Note that we simultaneously train both a conditional denoising network and an unconditional denoising network, so we randomly drop the goal point, language, and object color with a probability of 0.1. For the dropping operation, we set the goal point to be  $(-1e3, -1e3, -1e3)$ , the language to be null (''), and the object color to a gray image filled by 0.5.

**Vector field regression loss.** We include an auxiliary vector field regression loss  $\mathcal{L}_v$  to train goal predictor and contact predictor as introduced in Eqn. 9 and Eqn. 10. As introduced in Section C.1, we compute the ground-truth vector fields from the per-pixel direction pointing to the center of each cluster  $\hat{\mathbf{V}}^c$ , where  $c$  is the cluster index, the second rows in Fig. 13 and Fig. 12 provide examples of the ground-truth vector fields.

$$\mathcal{L}_v = \sum_c \|\hat{\mathbf{V}}^c - \mathbf{V}^c\|_2^2 \quad (16)$$

where  $\mathbf{V}^c$  are the predicted results from the contact predictor or the goal predictor for the vector field corresponding to the  $c$ -th cluster.

Section E provides more detailed ablation experiments to showcase the necessity of introducing this loss term for the goal predictor and the contact predictor.

### C.4. Inference Details

Here, we provide the inference details for each module within our affordance prediction pipeline.

**Open-set object detection.** We combine off-the-shelf GroundingDINO [56] and EfficientSAM [101] to acquire the 2D bounding box and foreground mask of the object of interest given language inputs, which will be used to acquire the masked object color image passed to the contact predictor and sample object points for the collision-avoidance guidance.

**Goal predictor inference.** Given the predicted goal probabilities from the goal predictor, we first select the pixel coordinates with top 5% predicted probabilities. We then fit a Gaussian model to the sampled pixel coordinates. We then sample 100 coordinates from the Gaussian model and

lift them to 3D using the predicted goal depth. These goal point samples will be used for  $\mathcal{J}_g$ .

**Contact predictor inference.** Given the predicted contact probabilities from the contact predictor, we first select the pixel coordinates with top 5% predicted probabilities. We fit a GMM model with  $k = 4$  to the sampled pixel coordinates. We then sample 100 coordinates from the GMM model and lift them to 3D using the queried depth from the cropped object depth. We additionally estimate the contact normal  $\mathbf{n}$  used for  $\mathcal{J}_n$  from the sampled contact points.

**Fine affordance predictor inference.** We leverage the classifier-free diffusion strategy as shown in Eqn. 15 to acquire the predictions. During the test time, we set  $w$  to be  $-0.7$  and generated 40 trajectories in parallel. For the strength of each guidance term, we set  $\lambda_g = 100$ ,  $\lambda_c = 200$ ,  $\lambda_n = 200$  for the articulated objects (e.g., cabinets, drawers, dishwashers), and change  $\lambda_n = 10$  for the portable objects (e.g., mugs, kettles, cans). For the collision-avoidance guidance  $\mathcal{J}_c$ , we sample 1024 points from the object surface if it’s portable and 1024 from the agent gripper. Hence  $N_p = 2048$  for the portable objects, and  $N_p = 1024$  for the articulated objects. The trajectories will be ranked using the final guidance cost  $\mathcal{J}$ . We further smooth the final trajectories using the Savitzky-Golay filter [82] for robot deployments.

### C.5. Robot Actions from Affordance

In the simulator, we use GraspNet [25] to initialize grasping poses and compute the end-effector poses such that the contact point lies between the fingers, then select the one with least collision with the scene. For pre-grasp actions, we use cuRobo library [88] to plan a path to reach the selected grasping pose and close the gripper upon reaching. During the interaction, we maintain the orientation constant and follow the interaction trajectory, whereby the simulator’s position controller uses Inverse Kinematics (IK) to compute joint configurations. For real robots, we follow similar principles, but use their respective API calls for pre-grasp action generation, and respective controllers. To enhance stability during real-world interaction trajectory following, the gripper orientation is adjusted to maintain a constant angle along the trajectory.

To control the two real robots, we treat Stretch 3’s base as one prismatic joint along an additional dimension when implementing our own IK solver to increase its reachability. Spot’s own whole-body controller adjusts its base while reaching the given end-effector pose for stability.

## D. Comparison to Recent Vision-Language-Action (VLA) Models

We add further discussion about recent representative VLA works: GR-2 [12],  $\pi_0$  [9] and RDT [57].

Teleoperated robot data is intrinsically directly actionable – but embodiment-specific and extremely labor-intensive to collect. In-the-wild human videos provide richer scene contexts but don’t possess actionable data. Accordingly, GR-2 [12] is pre-trained on human videos (and a few robot videos) by predicting future images rather than actionable trajectories, while  $\pi_0$  [9] and RDT [57] are pre-trained on teleoperated data only. Our key advantages are a scalable approach to extract agent-agnostic actionable information, i.e., the 3D affordance trajectories, from in-the-wild human videos, plus a learned affordance model, equipped with test-time guidance to enhance generalization. We additionally benchmark against it, by fine-tuning their model strictly following their provided recipe. The results are 72.1% v.s. 88.2% (Ours). We hypothesize 2 potential reasons for RDT: 1) not properly harnessing geometric cues from depth, while we exploit them for test-time guidance; 2) directly inferring actions from vision-language observations is challenging [59, 98, 102], but we address this with a hierarchical model.

## E. Additional Experiments

Here, we provide additional ablation experiments to validate the necessity of several architecture design strategies for our network.

### E.1. Real Robot Experiments

	RT01	RT02	RT03	RT04	RT05	RT06	RT07	RT08	RT09	RT10	RT11	Avg.
Ours	5/5	2/5	3/5	4/5	5/5	4/5	4/5	4/5	5/5	4/5	4/5	80.0

Table 4. Manipulation results on 11 real-world tasks across 2 robotic platforms evaluated on success rate (%). Each symbol denotes an ablation task: **RT01**: Pull drawer, **RT02**: Open cabinet, **RT03**: Take tissue, **RT04**: Drop paper ball, **RT05**: Close cupboard, **RT06**: Open right-side cabinet, **RT07**: Take bag, **RT08**: Push drawer, **RT09**: Close cabinet, **RT10**: Pick up toy, **RT11**: Press button. **RT01** - **RT07** are conducted by Hello Robot Stretch 3, and **RT08** - **RT11** are conducted by the Boston Dynamics Spot Robot.

The quantitative results of the real robot experiments are shown in Table. 4. These results further confirm the embodiment-agnostic nature and zero-shot transferability of our framework toward new scenarios.

### E.2. Ablation Studies on Trajectory Accuracy

We randomly sampled 120 video sequences from the HOI4D dataset [58] to analyze the impact of each module. Though more restricted to in-lab settings compared to the Epic Kitchens dataset [20] we used, this is the most similar dataset we know that provides GT hand box, mask, camera poses and depth to recover the GT affordance trajectory. We use the depth predictor, SfM system, hand-object detection



	S01	S02	S03	S04	S05
RMSE (m)	0.018	0.021	0.058	0.120	0.082

Table 5. Affordance trajectory accuracy. We ablate with 5 variants: **S01** all GT but hand box predicted, **S02** all GT but hand mask predicted, **S03** all GT but depth predicted, **S04** all predicted with scale optimization for SfM poses (Eqn. 1), **S05** adding pose refinement (Eqn. 2) to **S4**.

and segmentation models for each video to obtain the necessary data for trajectory extraction. Note that raw SfM poses cannot be used directly as they are scale-unaware. We ablate with 5 variants: **S01** all GT but hand box predicted, **S02** all GT but hand mask predicted, **S03** all GT but depth predicted, **S04** all predicted with scale optimization for SfM poses (Eqn. 1), **S05** adding pose refinement (Eqn. 2) to **S4**. The results are shown in Table. 5. The hand box prediction and mask segmentation have marginal impacts (**S1** and **S2**) as they are less involved in the reconstruction of the 3D hand trajectory. **S3** acts as our potential upper bound as it has GT camera poses. **S4** shows our optimization term recovers the metric scale. Using our formulated geometric constraints, **S5** further improves accuracy. We emphasize the importance of designing our affordance acquisition pipeline using RGB-only videos due to the abundance of such training data on the web. Note during deployment, our predicted trajectories are relative to accurate (sensed) depth; and test-time guidance using depth cues also increases the reliability.

### E.3. Ablation Studies on Goal Predictor

	AT01	AT02	AT03	AT04	AT05	AT06	Avg.
Ours [Full Model]	93.3	66.7	80.0	86.7	86.7	100.0	85.6
w/o vector field loss [GV1]	66.7	66.7	100.0	60.0	73.3	100.0	77.8
w/o box condition [GV2]	73.3	33.3	93.3	86.7	80.0	66.7	72.2
w/o obj condition [GV3]	80.0	66.7	80.0	86.7	80.0	53.3	74.4
w/o language condition [GV4]	33.3	26.7	73.3	66.7	86.7	53.3	56.7

Table 6. Ablation results for goal predictor design on 6 selected tasks evaluated on success rate (%). Each symbol denotes an ablation task: **AT01**: Close slide cabinet, **AT02**: Open hinge cabinet, **AT03**: Open microwave, **AT04**: Pull drawer, **AT05**: Open dishwasher, **AT06**: Pick up can from clutter.

As shown in Table. 6, we have four different model variants validated on the selected manipulation tasks for ablation studies. For **GV1**, we disable the vector field regression for the network, and thereby, it's trained without auxiliary vector field loss  $\mathcal{L}_v$ . For **GV2** - **GV3**, we drop different conditioning modalities, i.e., bounding box position features, object features, or language features.

We can first observe that the vector field as an auxiliary prediction can improve the success rate by 7.8% (**GV1**), demonstrating its effectiveness by "forcing" the goal predic-

tor to estimate more distinctive goal configurations that are most possible to execute. Both bounding box conditioning (**GV2**) and object feature conditioning (**GV3**) are useful to let the network yield more accurate goals, hence improving the task success rate through multi-goal guidance, which is the most important guidance term, as shown in Table. 6. Language conditioning is a critical factor in our approach. Without it, the predicted goal points often extend beyond the contact points, e.g., when given task instructions like 'pick up the cans,' the interaction trajectories conditioned on the inferred goal points are associated with placing actions. This mismatch leads to a drastic performance drop of nearly 30% (see **GV4**).

### E.4. Ablation Studies on Contact Predictor

	NSS↑	KLD↓	SIM↑
Ours [Full Model]	1.856	2.265	0.169
w/o vector field loss [CV1]	1.686	2.343	0.137
w/o language condition [CV2]	1.818	2.283	0.140

Table 7. Ablation results for contact predictor using samples from HANDAL dataset [33]. (↑ ↓: higher/lower is better.)

The ablation for the contact predictor is performed using commonly used metrics in the field for affordance prediction: Normalized Scanpath Saliency (NSS), Similarity (SIM), and Kullback-Leibler Divergence (KLD) to compare the distributions of the predicted contact regions and their ground truth [55, 56]. As shown in Table. 7, we validate model variants' performance on randomly selected samples from the HANDAL dataset [33]. Like ablation studies on goal predictor, **CV1** is a model trained without vector field loss, and **CV2** is a model trained without language conditioning.

From **CV1**, we see vector field loss is useful for contact affordance predictions. This is expected as we noticed some contact training samples could fail to inpaint human hands that occlude the contact regions. The vector field loss could mitigate the "occlusion" issue with dense per-pixel voting hints for the contact regions. The variant **CV2** also demonstrates the effectiveness of language conditioning.

### E.5. Ablation Studies on Fine Affordance Predictor

Though guidance provided by the cost functions during the test time has shown to be a crucial component in boosting the success of our trained fine affordance model, we still need to ensure the trained model can accurately capture the underlying distributions of the training data extracted from human videos. We hence set up a baseline with  $w = 0$  that generates trajectories without any guidance terms (**Ours** [No Guidance]). We verify the efficacy of conditioning the goal point (**TV1**), contact point (**TV2**), or the TSDF map (**TV3**), with results shown in Table. 8.

	<b>AT01</b>	<b>AT02</b>	<b>AT03</b>	<b>AT04</b>	<b>AT05</b>	<b>AT06</b>	<b>Avg.</b>
<b>Ours [No Guidance]</b>	<b>73.3</b>	<b>60.0</b>	<b>60.0</b>	<b>93.3</b>	26.7	<u>80.0</u>	<b>65.6</b>
w/o goal condition <b>[TV1]</b>	0.0	33.3	53.3	46.7	6.7	<b>100.0</b>	40.0
w/o contact condition <b>[TV2]</b>	<b>73.3</b>	40.0	73.3	<b>93.3</b>	<b>40.0</b>	60.0	<u>63.3</u>
w/o TSDF condition <b>[TV3]</b>	67.7	<b>60.0</b>	<b>60.0</b>	86.7	<u>33.3</u>	60.0	61.1

Table 8. Additional ablation results for fine affordance predictor design on 6 selected tasks evaluated on success rate (%). Each symbol denotes an ablation task: **AT01**: Close slide cabinet, **AT02**: Open hinge cabinet, **AT03**: Open microwave, **AT04**: Pull drawer, **AT05**: Open dishwasher, **AT06**: Pick up can from clutter.

As quantitatively shown by variant **TV1**, goal conditioning plays a crucial role in inferring plausible interaction trajectories with a performance decrease by 15.6%, again confirming the necessity of coarse affordance prediction as shown in Table. 2. Contact points conditioning (**TV2**) has less impact but is still helpful in improving performance. One subtle effect we observe is that the generated interactions from **TV2** has less awareness of collision avoidance. We explain it because no contact conditioning could lead to freely moving trajectories within the scene map without awareness of the absolute collision state of the trajectories. TSDF conditioning could implicitly provide scene collision cues to the fine affordance predictor, providing it bias to generate collision-free trajectories, whose effectiveness is again showcased particularly for the portable object manipulation task with an increase of 20.0% (**AT06**).

## E.6. Results on Robot Learning Applications

We provide detailed results of the visual goal-reaching and exploration tasks mentioned in Section. 4.4. As demonstrated in Fig. 9 and Fig. 10, our model outperforms competitors in most of the tasks.

## F. Additional Qualitative Results

Fig. 14 provides the contact prediction on samples from the HANDAL dataset [33]. Fig. 15 provides the predicted affordance by our method for in-the-wild RGB-D images and instructions. Fig. 16 provides an intuitive visual comparison illustrating the effect of each guidance term. Notably, collision-avoidance guidance enables the synthesis of safer collision-free trajectories, preventing the trajectory from colliding with the scene. Multi-goal guidance generates more accurate trajectories, particularly for objects like cabinets requiring circular interaction motions, while contact-normal guidance contributes to smoother trajectory generation.

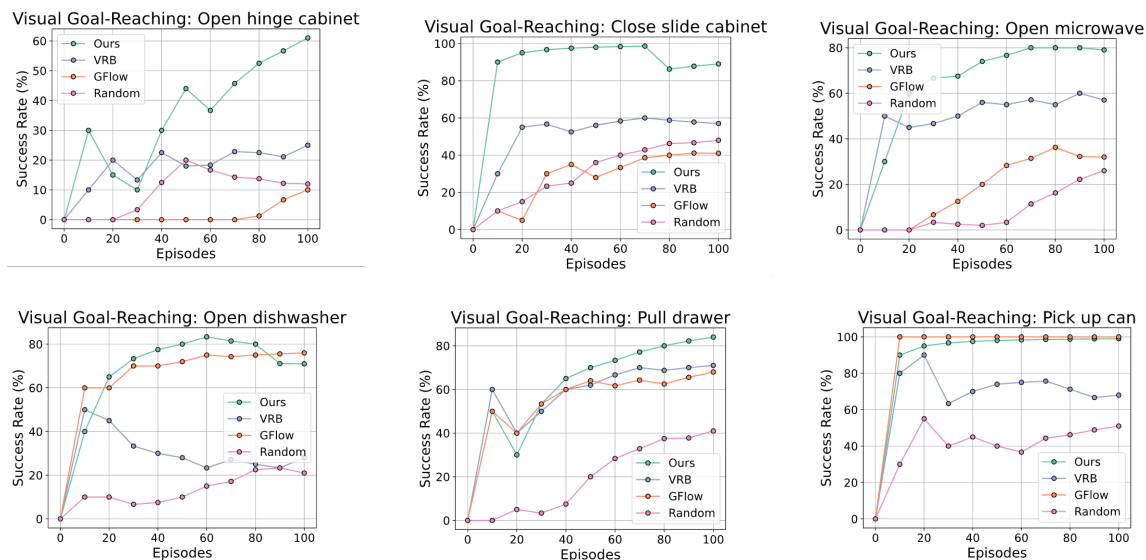


Figure 9. Detailed results of visual goal-reaching task.

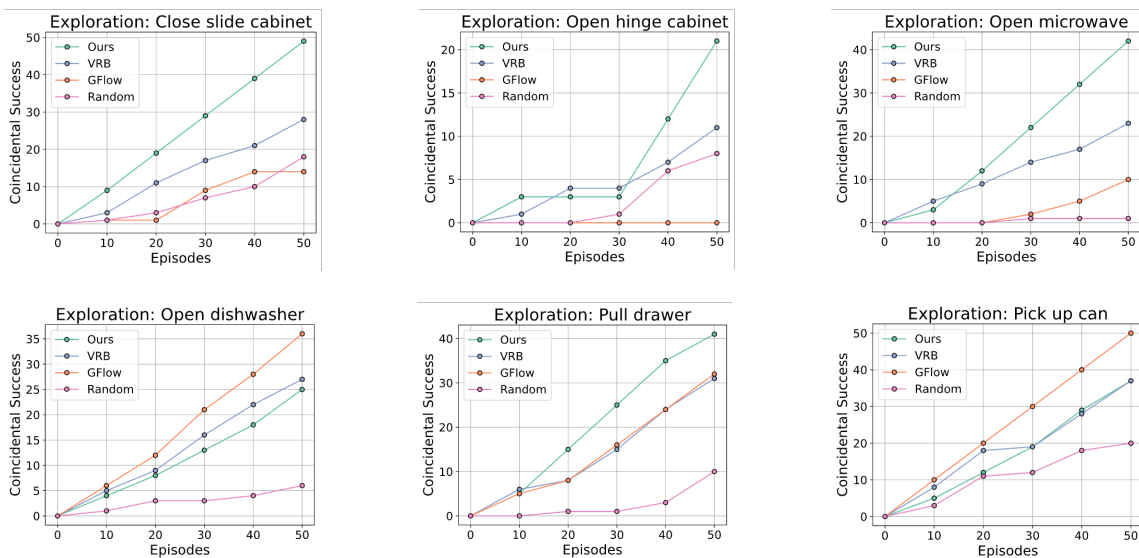


Figure 10. Detailed results of exploration task.

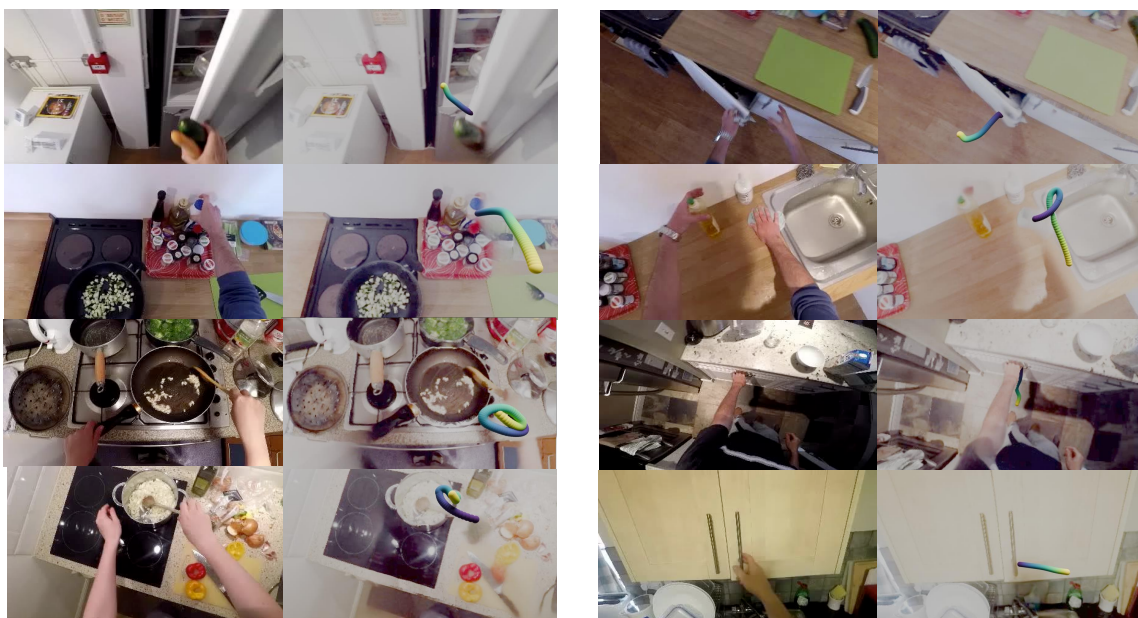


Figure 11. Example training samples to train fine affordance predictor.

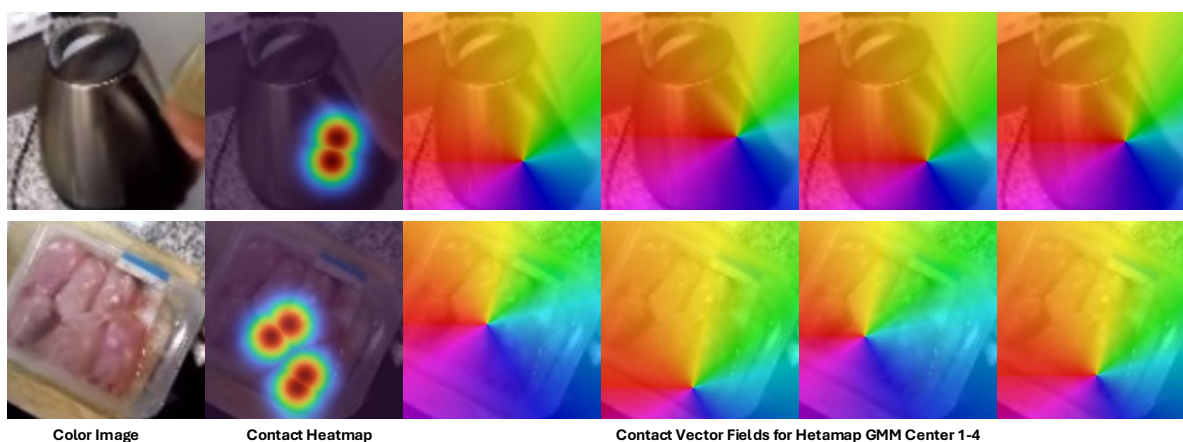


Figure 12. Example training samples to train contact predictor.



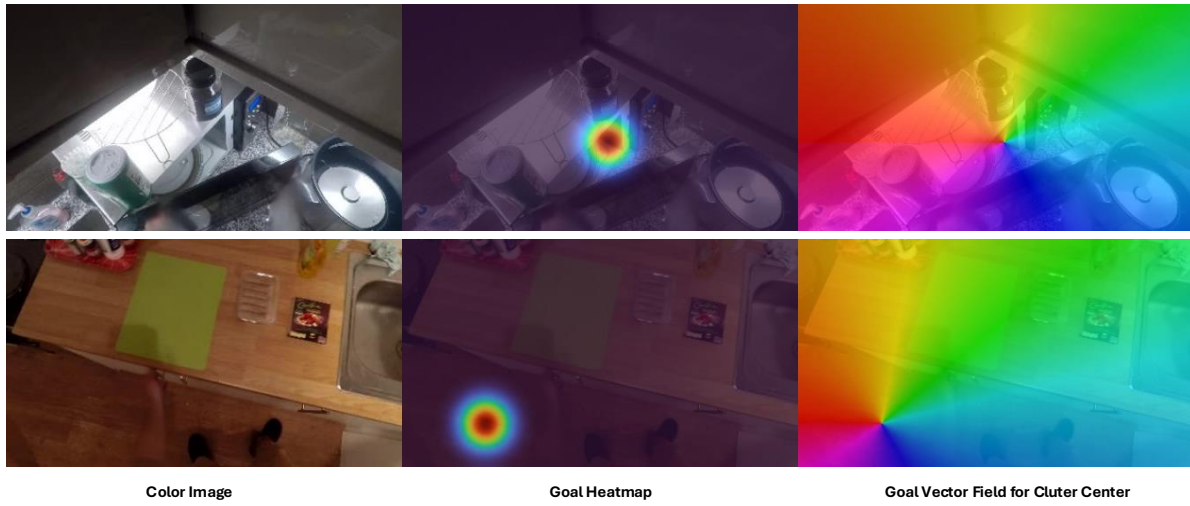


Figure 13. Example training samples to train goal predictor.

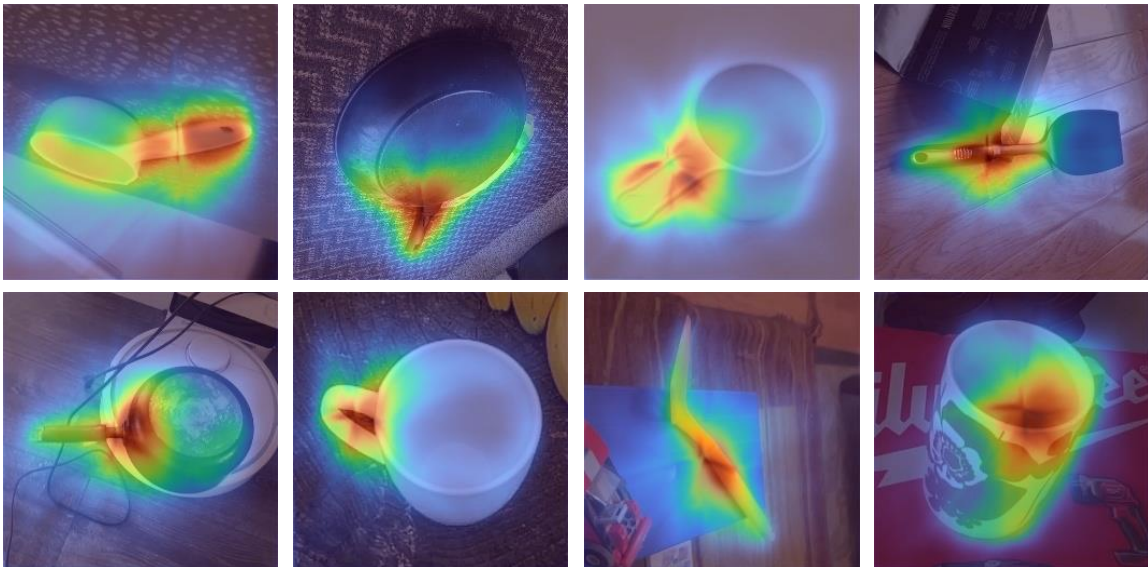


Figure 14. Contact predictions on samples from HANDAL dataset [33].

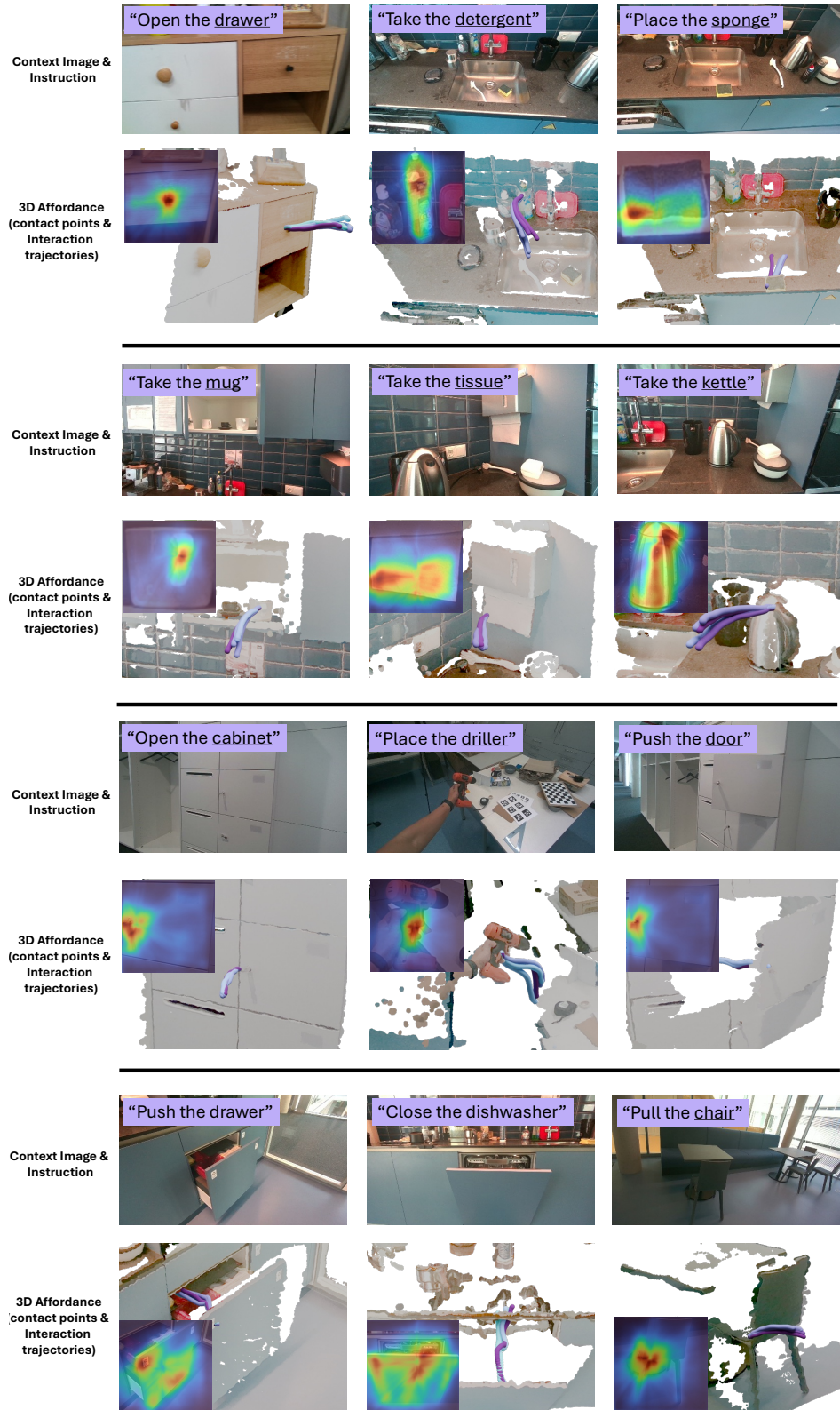


Figure 15. Our predicted 3D Affordance prediction (contact points and interaction trajectories) of in-the-wild data. For trajectories, darker color shades a lower final cost, yielding a higher rank for execution.

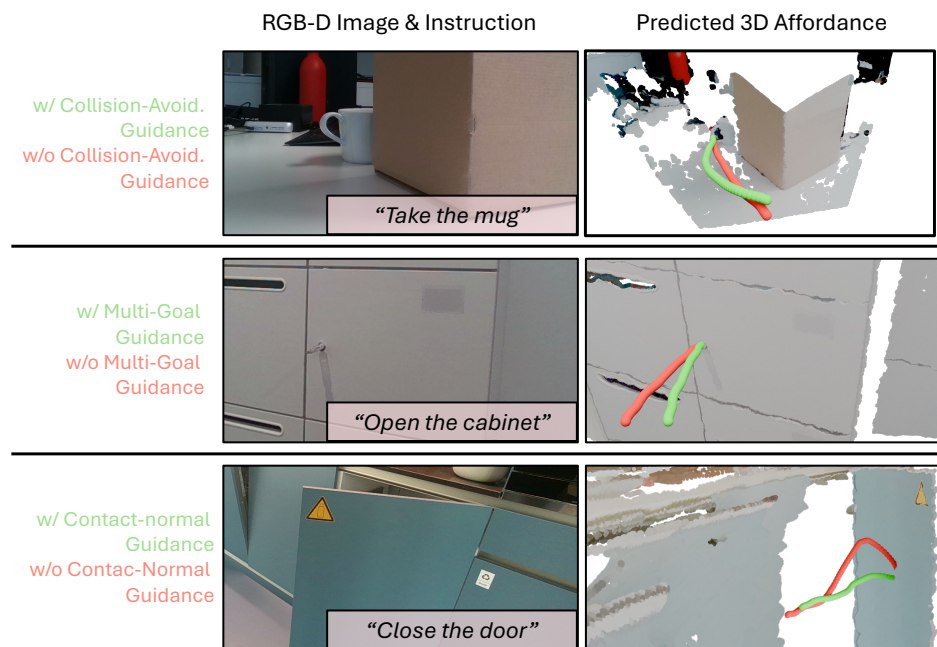


Figure 16. Qualitative comparison showcasing the impace of different guidance terms.

## References

- [1] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022. 3
- [2] Arpit Bahety, Priyanka Mandikal, Ben Abbatematto, and Roberto Martín-Martín. Screwmimic: Bimanual imitation from human videos with screw space projection. *arXiv preprint arXiv:2405.03666*, 2024. 2
- [3] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. *arXiv preprint arXiv:2207.09450*, 2022. 2, 3
- [4] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13778–13790, 2023. 2, 3, 6, 7, 8
- [5] Homanga Bharadhwaj, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Zero-shot robot manipulation from passive human videos. *arXiv preprint arXiv:2302.02011*, 2023. 3
- [6] Homanga Bharadhwaj, Abhinav Gupta, Vikash Kumar, and Shubham Tulsiani. Towards generalizable zero-shot manipulation via translating human interaction plans. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6904–6911. IEEE, 2024. 3
- [7] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables diverse zero-shot robot manipulation. *arXiv preprint arXiv:2405.01527*, 2024. 2
- [8] Shariq Farooq Bhat, Reiner Birkel, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 3
- [9] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al.  $\pi_0$ : A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 4
- [10] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 5, 3
- [11] Matthew Chang, Aditya Prakash, and Saurabh Gupta. Look ma, no hands! agent-environment factorization of egocentric videos. *Advances in Neural Information Processing Systems*, 36, 2024. 3
- [12] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024. 4
- [13] Annie S Chen, Suraj Nair, and Chelsea Finn. Learning generalizable robotic reward functions from “in-the-wild” human videos. *arXiv preprint arXiv:2103.16817*, 2021. 3
- [14] Hanzhi Chen, Binbin Xu, and Stefan Leutenegger. Funcgrasp: Learning object-centric neural grasp functions from single annotated example object. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1900–1906. IEEE, 2024. 2
- [15] Jiaqi Chen, Boyang Sun, Marc Pollefeys, and Hermann Blum. A 3d mixed reality interface for human-robot teaming. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11327–11333. IEEE, 2024. 1
- [16] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023. 3
- [17] Gene Chou, Yuval Bahat, and Felix Heide. Diffusion-sdf: Conditional generative modeling of signed distance functions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2262–2272, 2023. 3
- [18] Ching-Yao Chuang, Jiaman Li, Antonio Torralba, and Sanja Fidler. Learning to act properly: Predicting and explaining affordances from images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 975–983, 2018. 2
- [19] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European conference on computer vision (ECCV)*, pages 720–736, 2018. 7
- [20] 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. *International Journal of Computer Vision (IJCV)*, 130:33–55, 2022. 4, 1
- [21] Alexandros Delitzas, Ayca Takmaz, Federico Tombari, Robert Sumner, Marc Pollefeys, and Francis Engelmann. SceneFun3D: Fine-Grained Functionality and Affordance Understanding in 3D Scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2
- [22] Thanh-Toan Do, Anh Nguyen, and Ian Reid. Affordancenet: An end-to-end deep learning approach for object affordance detection. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 5882–5889. IEEE, 2018. 2
- [23] Bardienus Duisterhof, Lojze Zust, Philippe Weinzaepfel, Vincent Leroy, Yohann Cabon, and Jerome Revaud. Mast3r-sfm: a fully-integrated solution for unconstrained structure-from-motion. *arXiv preprint arXiv:2409.19152*, 2024. 8
- [24] Jakob Engel, Kiran Somasundaram, Michael Goesele, Albert Sun, Alexander Gamino, Andrew Turner, Arjang Talattof, Arnie Yuan, Bilal Souti, Brighid Meredith, et al. Project aria: A new tool for egocentric multi-modal ai research. *arXiv preprint arXiv:2308.13561*, 2023. 1



- [25] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11444–11453, 2020. 4
- [26] Daoyi Gao, Dávid Rozenberszki, Stefan Leutenegger, and Angela Dai. Diffcad: Weakly-supervised probabilistic cad model retrieval and alignment from an rgb image. *ACM Transactions on Graphics (TOG)*, 43(4):1–15, 2024. 3
- [27] Haoran Geng, Ziming Li, Yiran Geng, Jiayi Chen, Hao Dong, and He Wang. Partmanip: Learning cross-category generalizable part manipulation policy from point cloud observations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2978–2988, 2023. 2, 6
- [28] Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7081–7091, 2023. 6, 7
- [29] R Girshick. Fast r-cnn. *arXiv preprint arXiv:1504.08083*, 2015. 4
- [30] Mohit Goyal, Sahil Modi, Rishabh Goyal, and Saurabh Gupta. Human hands as probes for interactive object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3293–3303, 2022. 2
- [31] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19383–19400, 2024. 1
- [32] Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, et al. Maniskill2: A unified benchmark for generalizable manipulation skills. *arXiv preprint arXiv:2302.04659*, 2023. 6
- [33] Andrew Guo, Bowen Wen, Jianhe Yuan, Jonathan Tremblay, Stephen Tyree, Jeffrey Smith, and Stan Birchfield. HANDAL: A dataset of real-world manipulable object categories with pose annotations, affordances, and reconstructions. In *IROS*, 2023. 5, 6, 9
- [34] Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *arXiv preprint arXiv:1910.11956*, 2019. 6
- [35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2
- [36] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 1
- [37] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3, 4, 2
- [38] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022. 2, 3, 5
- [39] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 3
- [40] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021. 4, 2, 3
- [41] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022. 1
- [42] Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, 2022. 3, 4
- [43] Edward Johns. Coarse-to-fine imitation learning: Robot manipulation from a single demonstration. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 4613–4619. IEEE, 2021. 1
- [44] Bingyi Kang, Xiao Ma, Chao Du, Tianyu Pang, and Shuicheng Yan. Efficient diffusion policies for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024. 3
- [45] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. 2
- [46] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021. 3
- [47] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3
- [48] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 3, 1
- [49] Yuxuan Kuang, Junjie Ye, Haoran Geng, Jiageng Mao, Congyue Deng, Leonidas Guibas, He Wang, and Yue Wang. Ram: Retrieval-based affordance transfer for generalizable zero-shot robotic manipulation. *arXiv preprint arXiv:2407.04689*, 2024. 6, 7
- [50] Puhao Li, Tengyu Liu, Yuyang Li, Muzhi Han, Haoran Geng, Shu Wang, Yixin Zhu, Song-Chun Zhu, and Siyuan Huang. Ag2manip: Learning novel manipulation skills

- with agent-agnostic visual and action representations. *arXiv preprint arXiv:2404.17521*, 2024. 3, 6
- [51] 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*, pages 17562–17571, 2022. 3
- [52] Zhixuan Liang, Yao Mu, Mingyu Ding, Fei Ni, Masayoshi Tomizuka, and Ping Luo. Adaptdiffuser: Diffusion models as adaptive self-evolving planners. *arXiv preprint arXiv:2302.01877*, 2023. 3
- [53] Zhixuan Liang, Yao Mu, Hengbo Ma, Masayoshi Tomizuka, Mingyu Ding, and Ping Luo. Skilldiffuser: Interpretable hierarchical planning via skill abstractions in diffusion-based task execution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16467–16476, 2024. 3
- [54] Ziwei Liao, Binbin Xu, and Steven L Waslander. Toward general object-level mapping from sparse views with 3d diffusion priors. *arXiv preprint arXiv:2410.05514*, 2024. 3
- [55] Shaowei Liu, Subarna Tripathi, Somdeb Majumdar, and Xiaolong Wang. Joint hand motion and interaction hotspots prediction from egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3282–3292, 2022. 2, 3, 5
- [56] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 4, 3, 5
- [57] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024. 4
- [58] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21013–21022, 2022. 6, 4
- [59] Xiao Ma, Sumit Patidar, Iain Houghton, and Stephen James. Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18081–18090, 2024. 3, 4
- [60] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022. 2, 3
- [61] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021. 6
- [62] Kaichun Mo, Leonidas J Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6813–6823, 2021. 2, 6, 7
- [63] Austin Myers, Ching L Teo, Cornelia Fermüller, and Yiannis Aloimonos. Affordance detection of tool parts from geometric features. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1374–1381. IEEE, 2015. 2
- [64] Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. Grounded human-object interaction hotspots from video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8688–8697, 2019. 2
- [65] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022. 2, 3
- [66] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010. 3
- [67] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021. 1
- [68] Chuanruo Ning, Ruihai Wu, Haoran Lu, Kaichun Mo, and Hao Dong. Where2explore: Few-shot affordance learning for unseen novel categories of articulated objects. *Advances in Neural Information Processing Systems*, 36, 2024. 2
- [69] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024. 1, 6, 7
- [70] Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. 6
- [71] Georgios Papagiannis, Norman Di Palo, Pietro Vitiello, and Edward Johns. R+ x: Retrieval and execution from everyday human videos. *arXiv preprint arXiv:2407.12957*, 2024. 3, 1
- [72] Sebeom Park, Shokhrukh Bokijonov, and Yosoon Choi. Review of microsoft hololens applications over the past five years. *Applied sciences*, 11(16):7259, 2021. 1
- [73] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 523–540. Springer, 2020. 5
- [74] Dean A Pomerleau. Alvin: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988. 1

- [75] Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. In *European Conference on Computer Vision*, pages 570–587. Springer, 2022. 2, 3
- [76] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 4, 2
- [77] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, pages 416–426. PMLR, 2023. 2, 3
- [78] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022. 3
- [79] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. 1
- [80] Davis Rempe, Zhengyi Luo, Xue Bin Peng, Ye Yuan, Kris Kitani, Karsten Kreis, Sanja Fidler, and Or Litany. Trace and pace: Controllable pedestrian animation via guided trajectory diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13756–13766, 2023. 3, 5, 1
- [81] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 3
- [82] Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964. 2, 4
- [83] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3
- [84] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F Fouhey. Understanding human hands in contact at internet scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9869–9878, 2020. 3, 1
- [85] Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. Videodex: Learning dexterity from internet videos. In *Conference on Robot Learning*, pages 654–665. PMLR, 2023. 2, 3
- [86] Aravind Sivakumar, Kenneth Shaw, and Deepak Pathak. Robotic telekinesis: Learning a robotic hand imitator by watching humans on youtube. *arXiv preprint arXiv:2202.10448*, 2022. 3
- [87] Laura Smith, Nikita Dhawan, Marvin Zhang, Pieter Abbeel, and Sergey Levine. Avid: Learning multi-stage tasks via pixel-level translation of human videos. *arXiv preprint arXiv:1912.04443*, 2019. 3
- [88] Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, Nathan Ratliff, and Dieter Fox. curobo: Parallelized collision-free minimum-jerk robot motion generation, 2023. 4
- [89] Vadim Tschernezki, Ahmad Darkhalil, Zhifan Zhu, David Fouhey, Iro Larina, Diane Larlus, Dima Damen, and Andrea Vedaldi. EPIC Fields: Marrying 3D Geometry and Video Understanding. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2023. 4, 1
- [90] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 5
- [91] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023. 2, 3
- [92] Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and C Karen Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. *arXiv preprint arXiv:2403.07788*, 2024. 2, 3
- [93] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 8
- [94] Zan Wang, Yixin Chen, Baoxiong Jia, Puhao Li, Jinlu Zhang, Jingze Zhang, Tengyu Liu, Yixin Zhu, Wei Liang, and Siyuan Huang. Move as you say interact as you can: Language-guided human motion generation with scene affordance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 433–444, 2024. 3
- [95] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning, 2023. 3
- [96] Ruihai Wu, Yan Zhao, Kaichun Mo, Zizheng Guo, Yian Wang, Tianhao Wu, Qingnan Fan, Xuelin Chen, Leonidas Guibas, and Hao Dong. Vat-mart: Learning visual action trajectory proposals for manipulating 3d articulated objects. *arXiv preprint arXiv:2106.14440*, 2021. 2
- [97] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 3
- [98] Zhou Xian, Nikolaos Gkanatsios, Theophile Gervet, Tsung-Wei Ke, and Katerina Fragkiadaki. Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation. In *7th Annual Conference on Robot Learning*, 2023. 3, 4
- [99] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022. 2, 3
- [100] Haoyu Xiong, Quanzhou Li, Yun-Chun Chen, Homanga Bharadhwaj, Samarth Sinha, and Animesh Garg. Learning by watching: Physical imitation of manipulation skills

from human videos. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7827–7834. IEEE, 2021. [3](#)

- [101] Yunyang Xiong, Bala Varadarajan, Lemeng Wu, Xiaoyu Xiang, Fanyi Xiao, Chenchen Zhu, Xiaoliang Dai, Dilin Wang, Fei Sun, Forrest Iandola, et al. EfficientSAM: Leveraged masked image pretraining for efficient segment anything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16111–16121, 2024. [4](#), [3](#)
- [102] Mengda Xu, Zhenjia Xu, Cheng Chi, Manuela Veloso, and Shuran Song. Xskill: Cross embodiment skill discovery. In *Conference on Robot Learning*, pages 3536–3555. PMLR, 2023. [2](#), [3](#), [4](#)
- [103] Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. *arXiv preprint arXiv:2407.15208*, 2024. [3](#)
- [104] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jia Shi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381, 2024. [3](#), [4](#)
- [105] Jianglong Ye, Jiashun Wang, Binghao Huang, Yuzhe Qin, and Xiaolong Wang. Learning continuous grasping function with a dexterous hand from human demonstrations. *IEEE Robotics and Automation Letters*, 8(5):2882–2889, 2023. [3](#)
- [106] Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. *Conference on Robot Learning*, 2024. [2](#), [6](#), [7](#)
- [107] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*, 2024. [8](#)
- [108] Lingzhi Zhang, Shenghao Zhou, Simon Stent, and Jianbo Shi. Fine-grained egocentric hand-object segmentation: Dataset, model, and applications. In *European Conference on Computer Vision*, pages 127–145. Springer, 2022. [3](#), [1](#)
- [109] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 5628–5635. IEEE, 2018. [1](#)
- [110] Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. *Advances in Neural Information Processing Systems*, 36, 2024. [3](#)
- [111] Yi Zhou, Connelly Barnes, Lu Jingwan, Yang Jimei, and Li Hao. On the continuity of rotation representations in neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [2](#)
- [112] Yifeng Zhu, Arisrei Lim, Peter Stone, and Yuke Zhu. Vision-based manipulation from single human

video with open-world object graphs. *arXiv preprint arXiv:2405.20321*, 2024. [2](#)