How Do I Do That?
Synthesizing 3D Hand Motion and Contacts for Everyday Interactions

Aditya Prakash[*1]   Benjamin Lundell[2]   Dmitry Andreychuk[2]   David Forsyth[1]
Saurabh Gupta[†1]   Harpreet Sawhney[†2]
[1]University of Illinois Urbana-Champaign   [2]Microsoft
https://bit.ly/LatentAct
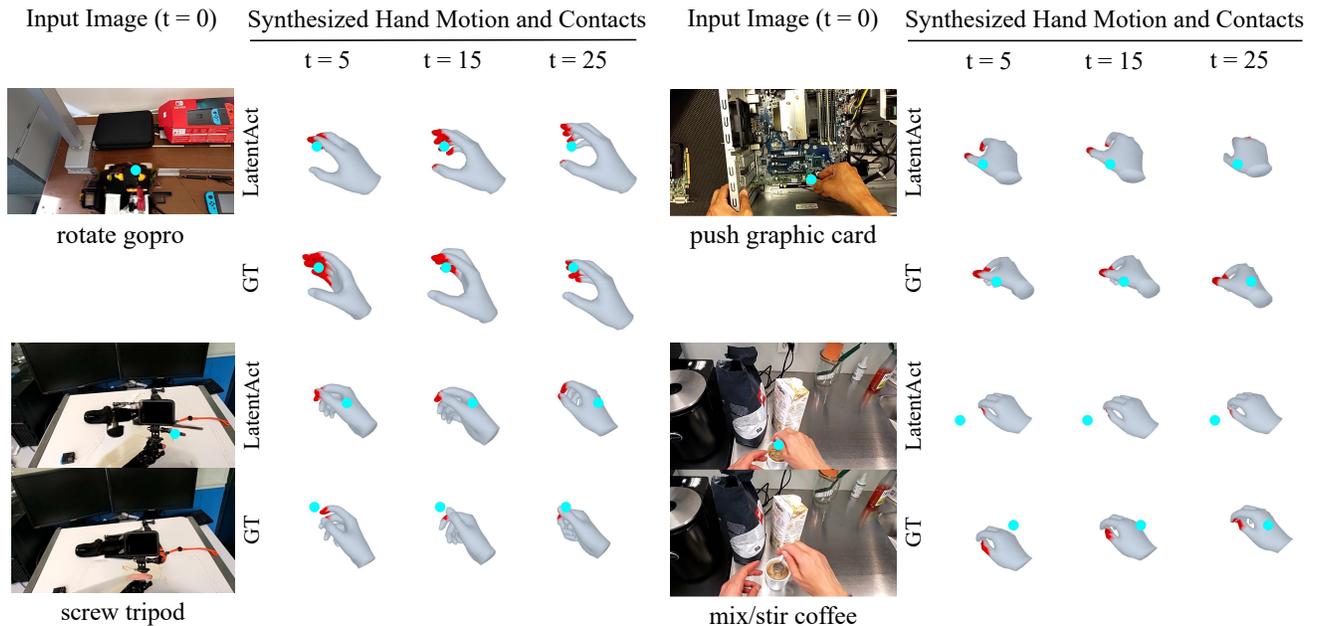
Figure 1. **Interaction Trajectories**. We tackle the novel task of predicting future 3D hand poses & contact maps, i.e., interaction trajectories, from a single image showing the object, text describing the action and a 3D contact point on the object, in everyday activites. We show the trajectory predicted by our method (LatentAct) & ground truth (GT) for 3 future timesteps along with the contact point. We consider 2 settings (top) Forecasting: single RGB view, action text & 3D contact points as input, (bottom) Interpolation: goal image is also provided.

Abstract

We tackle the novel problem of predicting 3D hand motion and contact maps (or Interaction Trajectories) given a single RGB view, action text, and a 3D contact point on the object as input. Our approach consists of (1) Interaction Codebook: a VQVAE model to learn a latent codebook of hand poses and contact points, effectively tokenizing interaction trajectories, (2) Interaction Predictor: a transformer-decoder module to predict the interaction trajectory from test time inputs by using an indexer module to retrieve a latent affordance from the learned codebook. To train our model, we develop a data engine that extracts 3D hand poses and contact trajectories from the diverse HoloAssist dataset. We evaluate our model on a benchmark that is 2.5-10× larger than existing works, in terms of diversity of objects and interactions observed, and test for generalization of the model across object categories, action categories, tasks, and scenes. Experimental results show the effectiveness of our approach over transformer & diffusion baselines across all settings.

1. Introduction

Hands interact with objects in diverse ways, often requiring varying skill levels to complete different tasks. Consider teaching someone to repair a bike. It is challenging to specify

---

what part to manipulate [41], what axis to flip a part over such that it inserts well, *etc*. Instead, it is easier to directly show how the hand should move and what parts of it should make contact with the object over time to complete the task. In this work, we study this problem. Given a single image of an object, a 3D contact point, and a text indicating the high-level action, we predict *Interaction Trajectories*, the sequence of 3D hand poses and contact maps that specify the initial and evolving hand interactions (Fig. 1), interpretable by a human. We target interaction with everyday objects that are often small, thin, transparent, occluded, and deformable (Fig. 1) for which 3D object models may not be available. Our work is a first step towards learning a motion prior for diverse interactions in everyday activities, that can be combined with 3D object models in the future.

In the absence of a 3D model, explicitly specifying contact points in the scene grounds the hand-object interaction, especially when there are multiple objects & 3D models are unavailable. This information is provided in the form of a 3D contact point. Moreover, we observe that while objects are diverse, hands interact with them in stylized ways [39]. There are only a few prototypical ways in which hands act in everyday interactions. To exploit this property, we learn a latent codebook of trajectories of 3D hand motions & contact points (Interaction Codebook, Sec. 3.1), effectively tokenizing *interaction trajectories* using a VQVAE [48]. Intuitively, the codebook learns the prototypical motion involved in (say) screwing in a screw that can be repurposed into motion involved in (say) closing a bottle cap. The codebook is indexed using visual inputs at test time to retrieve the closest codebook entry (Indexer module, Sec. 3.2), and aligned with the scene using the 3D contact point & visual context from the image input (transformer-decoder based Interaction Predictor, Sec. 3.3). We refer to our approach as *LatentAct*.

Since existing HOI datasets with 3D annotations have constrained setups, we use egocentric videos from the HoloAssist dataset, showing everyday interactions. We leverage recent advances in 3D hand pose estimation [30] and segmentation in videos [36] to build a semi-automatic data engine (Sec. 3.4) to extract trajectories of 3D hand poses and contact maps from HoloAssist for 800 tasks performed by ordinary users across 120 object categories and 24 action categories. Our setup is 2.5X-10X larger than existing works, *e.g*. HOT3D [2] (33 rigid objects), ARCTIC [11] (10 articulated objects), GRAB [43] (51 objects), HOI4D [28] (16 object categories), in terms of diversity of objects and interactions observed in everyday activities.

For evaluation, we focus on generalization to 4 aspects: novel object categories, action categories, tasks and scenes. We show results on 2 settings: (1) Forecasting: the model generated predictions conditioned on the textual description of the action, current image showing the object of interaction and the 3D contact point at the current timestep. (2) Inter-

polation: Here, we also provide the goal image, along with the previous inputs, showing the final state of interaction. For each of these settings, we consider 2 variants: (a) hand visible in the image, (b) hand is absent in the image (often the case at start of interactions). Across this large-scale experiment setting, we find our approach to generalize better than transformer & diffusion baselines that directly learn to predict interaction trajectories from images (Sec. 4.2).

2. Related Work

Generating HOI sequences: Recent works in HOI [3, 7, 13, 26, 44, 54, 55] have studied generating hand-object motions in different settings, e.g. conditioned on textual descriptions [7, 13], body poses [3, 13, 26, 44] for both rigid [3, 7, 13, 26] and articulated objects [4, 55]. While they focus on pick & place tasks involving simple & constrained interactions, we consider everyday activities, often involving small, thin, textureless, occluded, articulated, deformable and dynamic objects in contact for which obtaining 3D models is non-trivial. Existing works in this domain use either object-centric [4, 7] or hand-centric [44, 54] frame to represent trajectories, this becomes a limitation in our setting since 3D objects are unknown and hands may not be present in the image (often the case at the start of the interaction). Instead, we focus on predicting future 3D hand poses & contact maps from a single image showing the object, text describing the action & a 3D contact point to localize the interaction, for diverse interactions in everyday activities.

HOI datasets: Existing works use datasets [11, 17, 18, 20, 24, 28] collected in controlled settings which provide 3D ground truth using MoCap [11, 43] or multi-camera setups [17, 18, 20, 24, 28, 29]. However, the diversity of objects and interactions in these datasets is limited due to constrained nature of the capture setup. While recent datasets [6, 9, 40] have explored natural interactions in egocentric videos [8, 15], they only provide 2D annotations, *e.g*. segmentation masks [6, 9], 2D bounding boxes [40] and grasp labels [6]. To mitigate this issue, we design a data engine to extract 3D hand poses and contact maps (Sec. 3.4) from egocentric videos showing everyday interactions. Specifically, we extract 3D trajectories from HoloAssist for 800 tasks performed by ordinary users across 120 object categories and 24 action categories, which is 2.5X-10X larger than existing datasets [2, 11, 28, 43].

HOI representation: Hands are often represented using a parametric mesh model, e.g., MANO [37], along with the global translation and rotation of the wrist [7, 11, 13, 30, 31, 33, 51, 53, 54]. Given an object mesh, its 6DoF pose can be denoted using translation & rotation [13, 55], signed distance field [54] or using Basis Point Set (BPS [34]) distances [7, 26]. For hand-object contact, existing approaches typically consider a canonicalized representation with the hand [4, 54] or object [7] at the origin & use relative dis-
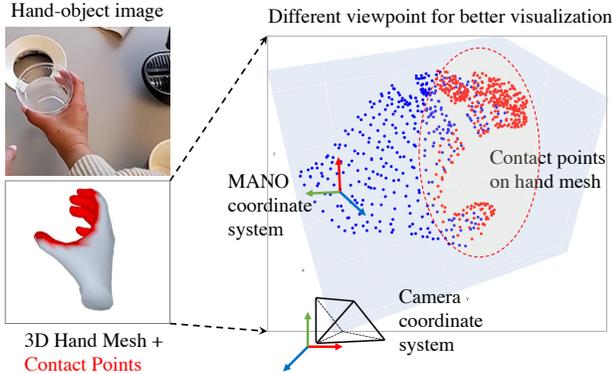
Figure 2. We represent contact points as binary masks on the hand mesh vertices. The hand mesh is represented in the camera coordinate system, consisting of the local hand pose in the MANO [38] coordinate frame and a global transformation from the MANO frame to the camera frame. These hand poses and contact points over several timesteps form interaction trajectories.

tances [7, 11, 12, 32, 44] or a distance field [54] from hand joints to represent contact. Since 3D hand-object motion annotations are unavailable for diverse objects, e.g., thin, occluded, deformable, we instead focus on sequences of 3D hand poses and contact maps, *i.e.*, interaction trajectories.

Affordances: Gibson [14] developed the concept of affordances as a set of functionalities that the environment furnishes to an agent, which can be a human, a robot, an animal, or hands. Affordances are typically represented as the contact regions on the objects [1, 5, 23, 25, 27, 28, 43, 50] & the agent [1, 19, 23, 42, 43, 47, 53, 53, 54]. They have been explored in domains like robotics [1, 42], human interactions [19, 43, 47, 49] & hand grasp synthesis [23, 42, 43, 53, 54]. Recent works have also proposed generative models for affordances synthesis [49, 53, 54].

3. Method

Given a single RGB view, action text, and a 3D contact point as input, we focus on predicting 3D hand poses & contact points, *i.e.*, interaction trajectories (Fig. 1), for future timesteps. Our approach consists of: (1) Interaction Codebook: learns a latent codebook of affordances using a VQVAE (Fig. 3a), (2) Learned Indexer: maps the action text, image (showing an object) & contact point inputs to the codebook indices to obtain the corresponding embeddings (Fig. 3b), (3) Interaction Predictor: takes the queried embeddings along with text, image and contact point inputs to output the 3D interaction trajectory (Fig. 3b).

Hand mesh: We use the MANO [37] model to represent the hand. It consists of pose θ and shape β parameters, which can be converted to a 3D hand mesh. These mesh vertices lie in the MANO coordinate system, which can be transformed to the *camera coordinate system at the first*

timestep (reference frame) using a rotation and translation. This gives us the hand trajectory in the reference frame.

Contact map: We represent the contact points as a binary mask over the MANO hand mesh vertices, *i.e.*, a 778 dimensional binary vector for each timestep, referred to as contact map in the paper (Fig. 2). The contact points may also change over time depending on the action, *e.g.* twisting.

Once the 3D hand mesh is transformed into reference frame, we estimate the trajectory of the contact points by computing the centroid of the contact points at each timestep. The contact centroid in the reference frame is used as the 3D contact point input during training. We assume a 3D contact point to be available in the first frame at test time. To input this point, we create a 3D voxel grid of resolution $16 \times 16 \times 16$. We take minimum and maximum values of 3D hand trajectory (along each axis, in the reference frame) across all training sequences. This span is divided into 16 equal parts per axis, associating each voxel to 3D metric range. It is kept fixed at train & test time. Thus, the center of each voxel represents a 3D location in metric space, *i.e.* a 3D metric grid, in the camera coordinate system. We create a 3D gaussian heatmap centered at 3D contact point in this grid, referred to as voxelized contact points in the paper. This changes as the trajectory evolves. This is different than the contact map (binary mask over mesh vertices).
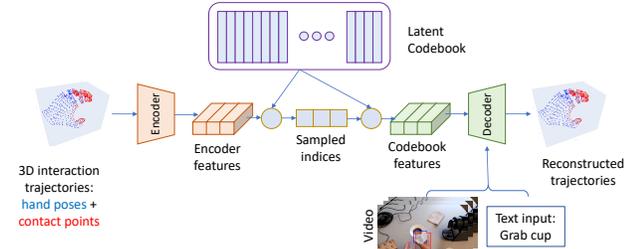
3.1. Interaction Codebook (InterCode)

This is implemented as a VQVAE with a transformer encoder-decoder architecture [16]. It consists of 3 modules:

Encoder: The hand and contact map trajectories for a fixed time horizon T are passed to a transformer encoder to get the feature embedding. These features are used to query the codebook to obtain the relevant embeddings using L2 distance. Specifically, the input trajectory, consisting of MANO parameters & contact maps, is first passed to a 1-layer MLP to match the feature dimension of the transformer. We use a 1-layer transformer encoder with 1 head & feature dimension = 512. It outputs a $T \times 512$ dimensional feature.

Latent codebook: It is initialized as a $K \times E$ dimensional matrix and is updated during training using exponential moving average of the embeddings output from the encoder, following recent works [16]. Concretely, we use a multi-layer residual VQVAE architecture [16]. We set the number of quantizers for VQVAE to 6, $K = 512$ and $E = 512$. When querying the codebook during training, sampling is done using the Gumbel-SoftMax trick [21] to preserve differentiability. We set the temperature of Gumbel-Softmax to 0.5 & inject noise at training for stochastic sampling. At test time, argmax is used to get the closest codebook entry.

Decoder: It takes multiple inputs: (1) the latent embeddings queried from the codebook ($T \times 512$), (2) CLIP [35] embeddings of the text describing the action (512 dimensional feature extended for T timesteps), (3) DeiT [46] embeddings

(a) Interaction Codebook is a VQVAE model that learns a latent codebook of 3D interaction trajectories, consisting of a transformer encoder-decoder architecture. The encoder features are used to sample codebook indices and corresponding embeddings which are passed to a decoder that also takes in the video input and text describing the action and reconstructs the 3D interaction trajectories.
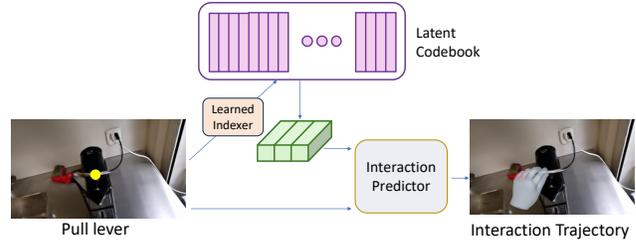
(b) We train an Indexer module to map the inputs at test time to the codebook indices to extract the relevant embeddings. These are then passed to a interaction predictor module that outputs the 3D interaction trajectories. We consider 2 settings: (1) Forecasting: text describing the action, single image & 3D contact point, (2) Interpolation: also providing the goal image showing the final state of the interaction (not shown here for clarity).

Figure 3. **Overview**. Our framework involves a 2-stage training procedure: (left) Interaction Codebook: to learn a latent codebook of hand poses and contact points, *i.e.*, tokenizing interaction trajectories, (right) a learned Indexer & an Interaction Predictor module to predict the interaction trajectories from single image, action text & 3D contact point. We use pretrained features for images (from DeiT [46]) and text (from CLIP [35]). 3D contact point is input as a 3D gaussian heatmap in a 3D voxel grid (omitted here for clarity).

of the video showing the actions ($T \times 768$), (4) voxelized contact points at each timestep, which are passed through four 3D convolution layers. We also pass in a 4D grid with each voxel containing the 3D location of its center in metric space, and process with 3D convolution layers. This gives us a 32 dimensional feature vector from the contact module. Concatenating all these input features, we get a 1824-dimensional joint feature that is passed through a 1-layer MLP to change the dimensionality to 512 and then fed to a transformer decoder (1-layer with 1 head & dropout of 0.2). The query embeddings for the transformer decoder are set as trainable parameters. It outputs a $T \times 512$ feature map which is passed to separate decoders for predicting the MANO parameters and the contact map. We use a 2-layer MLP to predict the MANO parameters and a 3-layer MLP to predict the contact maps for the entire time horizon T.

Training: The loss function consists of several terms: (1) Smooth L1 loss on the MANO parameters, (2) L1 loss on the contact centroid, (3) L1 loss on the translation between the future frame and first frame, (4) L1 loss on the 6D rotation representation between the future frame and first frame, (5) binary cross-entropy loss on the contact map predictions.

3.2. Learned Indexer

At test time, we need to query the learned codebook to get the latent interaction embeddings. This is done using a learned module which takes the test time inputs, i.e. text describing an action, image showing an object & a 3D contact point, and outputs a probability distribution over the codebook indices. The indices and corresponding embeddings can be sampled using the predicted probabilities. Note that the test time inputs only consist of a single image and a 3D contact point, unlike the entire trajectory used for training InterCode.

Specifically, we first compute features from different inputs: CLIP embedding of text (512), DeiT embeddings of

images (768), features from voxelized contact point, in the reference frame, after passing through 4 3D convolutional layers (32) and the 3D contact point. We concatenate these features and process through a 2-layer MLP (1024 & 512 nodes) to match the input dimensions of the transformer classifier module. It is implemented as a 1 layer, 1 head transformer decoder layer that takes in learnable query embeddings and aforementioned features from different modalities to output $T \times 512$ dimensional features. These features are then passed to a 3-layer MLP to output a probability distribution over the indices of the codebook.

Training: It consists of a cross-entropy loss with ground truth computed as the closest codebook indices (L2 distance) from the encoder features of the InterCode module.

3.3. Interaction Predictor (InterPred)

Since the inputs at test time differ from the decoder inputs of the InterCode module during training, we need a separate learned module to predict the interaction trajectories. It consists of a transformer decoder module with the following inputs: (1) the latent embeddings queried from the codebook using the learned prior, (2) CLIP embeddings of the text describing the action, (3) DeiT embeddings of the image showing the object, and (4) voxelized contact point in the reference frame. These inputs are processed in the same way as for Learned Indexer before passing to the transformer decoder. The architecture is same as the decoder in Inter-Code. While the inputs are different than InterCode, they are passed to an MLP to match transformer dimension.

Training: The loss function consists of several terms: (1) Smooth L1 loss on the MANO parameters, (2) L1 loss on the contact centroid, (3) L1 loss on the translation between the future frame and first frame, (4) L1 loss on the 6D rotation representation between the future frame and first frame, (5) binary cross-entropy loss on the contact map predictions.
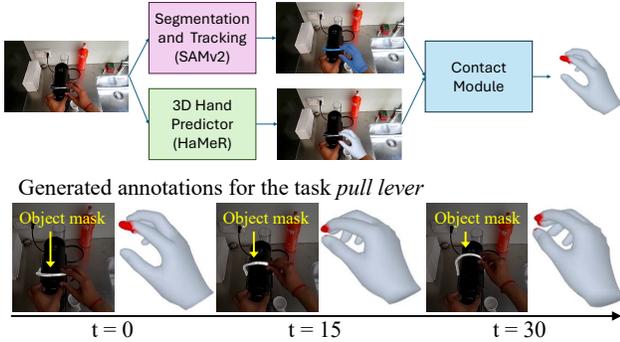
Figure 4. **Data engine**. (*top*) Object masks are extracted using SAMv2 [36], 3D hand poses & masks (2D rendering of mesh) are from HaMeR [30], contact points are computed by projecting the 3D hand points into the 2D contact region (intersection of hand & object masks). (*bottom*) Generated object masks (highlighted in white), 3D hand mesh & contact points for 3 timesteps.

3.4. Data Engine

Existing lab datasets are limited to simple interactions in constrained setups. Instead, we use egocentric videos from HoloAssist showing everyday interactions with rigid, articulated & deformable objects, *e.g.* operating machines, tool use, kitchen tasks. It consists of 2k+ videos with temporally localized atomic actions, e.g. *grab, open, screw, mix, rotate, align, slide,* etc. with diverse objects. We extract these clips of atomic actions and convert them into the required format for training & evaluation. We consider 24 action categories with 120 object categories spanning 800 tasks. This is 2.5-10× larger than existing datasets, *e.g.* HOT3D [2] (33 rigid objects), ARCTIC [11] (10 articulated objects), GRAB [43] (51 objects), HOI4D [28] (16 object categories), in terms of objects & diversity of interactions. While HoloAssist contains diverse interactions, it does not provide 3D annotations for hand poses & contact points. To extract these annotations, we design a semi-automatic data pipeline (Fig. 4) using 2D segmentation masks, 3D hand poses & 2D contact region.

Segmentation masks: We adopt a semi-automatic approach for object masks, that involves clicking a point on the target object in the first frame of a video & tracking the object mask across the video using SAMv2 [36]. For 2D hand masks, we render the 3D hand mesh (described next) into the image.

3D hand poses: We use the off-the-shelf HaMeR [30] model to get 3D hand meshes in each video frame. HaMeR takes an image crop around the hand as input (estimted from Hands23 model [6]). The predicted 3D mesh in each frame is transformed to the reference frame using known intrinsics & extrinsics available in the HoloAssist dataset.

3D Contact points: We first compute a 2D contact region in an image as the overlap between the hand & object masks. We add gaussian noise at the boundary to obtain a dense 2D contact region. We back project the 2D contact region into 3D hand mesh to obtain 3D contact points on the hand mesh.

4. Experiments

4.1. Protocol

Task: We predict trajectories of MANO hand parameters & contact map (a binary mask over 778 vertices of the MANO [38] hand mesh) in 2 settings: (1) Forecasting: input consists of a textual description of the action, a current image showing the object of interaction, and a 3D contact point. (2) Interpolation: We provide the goal image, along with the previous inputs, showing the final state of interaction. For each setting, we 2 two variants: (a) current image with a hand, (b) current image without a hand (removed via in-painting [52]) since the hand may not be visible before the start of the interaction in many practical scenarios. We only consider right hand motion in this work.

Generalization aspects: We study generalization capabilities for 4 different settings: (1) Object categories, (2) Action categories, (3) Novel tasks: a combination of object & action categories, (4) Novel scene: we hold out videos from 1 location (data is collected in 2 locations). For each setting, we create train, validation & test splits in the ratio 80:10:10.

Metrics: (1) MPJPE: Mean Per Joint Position Error between predicted & ground truth hand joints, averaged over all future timesteps. (2) MPJPE-PA is the procrustus aligned variant, which optimizes for a single rotation, translation & scale to align the entire trajectory with ground truth trajectory, before computing MPJPE. (3) F1 score: harmonic mean of precision & recall to measure accuracy of contact maps .

Baselines: Since no prior works predict interaction trajectories from image input, we adapt 2 recent baselines from human pose literature to work in our setting: (1) HCTFormer: The input image is encoded using a ViT [10] encoder followed by a transformer decoder, similar to the pose decoder used in T2P [22]. TCP decoder takes in trajectory query (for global intent), pose query & pose embeddings. In our case, intent comes from text. We replace pose embeddings with {image, contact, text} features, pass them to an MLP & then feed to {TRM, MLP} modules, as in TCP. (2) HCTDiff: We modify the motion diffusion model (MDM [45]) for hand trajectory and contact map prediction. For this, we change conditioning to use {image, text, contact} features but diffusion architecture is same. Both HCTFormer and HCTDiff are trained in the same setting as ours for fair comparisons.

Existing methods that use object-centric representations [4, 7] can not be applied in our setting since we do not have the 3D object models. Hand-centric representations [44, 54] are also not suitable in our setting since the hand is not always visible in the input image at test time.

Our approach involves a 2-stage training procedure: InterCode followed by Indexer & InterPred, from different inputs. We train with both transformer & diffusion variants for InterPred, referred to as LatentAct & LatentAct-Diff respectively. LatentAct-Diff uses same InterCode & Indexer as

	Method	Task-level			Object-level			Action-level			Scene-level		
		M-PE(cm)↓	M-PA(cm)↓	F1↑	M-PE(cm)↓	M-PA(cm)↓	F1↑	M-PE(cm)↓	M-PA(cm)↓	F1↑	M-PE(cm)↓	M-PA(cm)↓	F1↑
Hand visible	**Forecasting**												
	HCTFormer	8.32	3.06	0.72	8.70	3.19	0.72	8.59	3.25	0.73	8.52	3.09	0.72
	HCTDiff	8.42	**2.75**	0.72	9.01	**2.88**	0.73	9.11	**3.10**	0.74	9.49	**2.84**	0.71
	LatentAct (Ours)	**7.61**	2.99	**0.75**	**8.09**	3.06	**0.75**	**8.14**	3.32	**0.77**	**7.87**	3.04	**0.74**
	Interpolation												
	HCTFormer	7.52	3.02	0.76	8.10	3.10	0.76	7.77	3.29	0.78	7.54	3.00	0.76
	HCTDiff	8.30	**2.72**	0.77	8.52	**2.91**	0.78	8.38	**3.16**	0.78	8.60	**2.81**	0.77
	LatentAct (Ours)	**6.72**	2.87	**0.80**	**7.48**	3.02	0.78	**7.25**	3.19	**0.80**	**7.27**	2.93	**0.78**
No hands	**Forecasting**												
	HCTFormer	8.34	2.95	0.72	8.85	3.02	0.73	8.54	3.21	0.74	8.53	3.05	0.72
	HCTDiff	8.85	**2.65**	0.72	9.74	**2.81**	0.72	9.50	**3.11**	0.74	9.21	**2.74**	0.71
	LatentAct (Ours)	**7.93**	2.97	**0.76**	**8.26**	3.01	**0.76**	**8.19**	3.28	**0.77**	**7.90**	3.00	**0.75**
	Interpolation												
	HCTFormer	7.30	2.95	0.76	8.42	3.15	0.76	8.01	3.30	0.77	7.29	2.99	0.77
	HCTDiff	8.27	**2.76**	0.76	8.68	**2.91**	0.76	9.33	**3.10**	0.78	10.02	**2.82**	0.76
	LatentAct (Ours)	**7.13**	2.89	**0.79**	**7.24**	2.97	**0.79**	**7.44**	3.21	**0.80**	**7.12**	2.89	**0.78**

Table 1. **Generalization results**. We report MPJPE (M-PE: cm), MPJPE-PA (M-PA: cm) & F1-score in 4 generalization settings: novel tasks, objects, actions & scene, to measure the accuracy of the predicted trajectories. We adapt two recent methods from human pose literature to work with image inputs, *i.e.*, HCTFormer: ViT encoded image features passed to a transformer decoder (similar to pose decoder of T2P [22]), HCTDiff: MDM [45] modified to take image features as well. LatentAct leads to better absolute hand poses & contact maps.

LatentAct but different InterPred, *i.e.*, an iterative denoising model from MDM instead of a single step prediction. The denoising is done in latent space of the codebook followed by a decoder to generate trajectories. It is trained using L2 loss on denoised embedding and loss in Sec. 3.3.

4.2. Results

Generalization results: First, we study if the model can generate trajectories from the text describing the action, single image & 3D contact point, *i.e.*, Forecasting task. In the absence of a goal, this is quite challenging and inherently multimodal in nature. We compare with the GT trajectories for all 4 generalization aspects: novel objects, actions, tasks & scene. We report MPJPE, MPJPE-PA for hand trajectories & F1-score for contact maps. We observe that our approach leads to consistent gains in absolute hand poses & contact maps. We show results (Tab. 1) on 2 variants of the task, (1) hand visible in the image, (2) hand absent in the image, often the case at the start of interactions. The hand visible in the image makes the tasks slightly easier since the initial pose is visible, as also evident from the results.

Next, we also provide the goal image, along with the previous inputs, showing the final state of interaction, referred to as goal-conditioned interpolation. This is slightly easier than the Forecasting task since the final state of the interaction is also provided. The results (Tab. 1) on 2 varaints of the task, (1) hand visible in the image, (2) hand absent in the image, show similar benefits as in Forecasting.

These results are for T=30 timesteps. We also report results for T=16 in the supplementary with similar trends.
Results on ARCTIC: We also show results (Tab. 2) on ARCTIC [11] to check if LatentAct leads to gains on existing

Method	MPJPE↓	MPJPE-PA↓	F1↑
HCTFormer: trained on ARCTIC	15.76	3.61	0.36
LatentAct: trained on ARCTIC	**14.77**	3.76	0.41
LatentAct: zero-shot from Holo	15.72	3.71	0.19
LatentAct: Codebook & Indexer from Holo	15.36	**3.58**	**0.45**

Table 2. HoloAssist is significantly larger than ARCTIC in terms of contact sequences leading to benefits in zero-shot generalization & transferring models trained on HoloAssist to ARCTIC.

lab datasets. ARCTIC has accurate 3D labels but is limited in scale ($10\times$ smaller than HoloAssist for contact sequences). When trained only on ARCTIC, we find the trends to transfer from HoloAssist, *i.e.*, LatentAct is better than HCTFormer. Using codebook & indexer trained on HoloAssist further benefits models trained on ARCTIC. Zero-shot transfer from HoloAssist is competitive to models trained on ARCTIC.
Benefits of Interaction Codebook: We verify if the 2-stage training procedure is beneficial by comparing with single-stage training methods HCTFormer & HCTDiff in Tab. 4 and observe consistent benefits. Another option is to retrieve an interaction trajectory from the training set, conditioned on the input. However, this does not scale well with the training data & is computationally expensive, *e.g.* the learned codebook in InterCode has 512 entries whereas the training dataset has ∼15K sequences, which is significantly larger.
Contact maps helps hand predictions: Here, we train the InterCode & InterPred modules with only the hand pose loss, i.e., removing the loss on contact maps. In Tab. 5, we observe that loss on contact maps helps with hand predictions.
Trends with LatentAct-Diff:. Training InterPred with diffusion losses leads to better hand poses but worse contact maps compared to LatentAct, across all settings (Tab. 3).

	Method	Task-level			Object-level			Action-level			Scene-level		
		M-PE(cm)↓	M-PA(cm)↓	F1↑	M-PE(cm)↓	M-PA(cm)↓	F1↑	M-PE(cm)↓	M-PA(cm)↓	F1↑	M-PE(cm)↓	M-PA(cm)↓	F1↑
Hand visible	**Forecasting**												
	LatentAct	7.61	2.99	0.75	8.09	3.06	0.75	8.14	3.32	0.77	7.87	3.04	0.74
	LatentAct-Diff	7.90	3.03	0.71	8.49	3.20	0.72	8.29	3.34	0.72	8.32	3.10	0.72
	Interpolation												
	LatentAct	6.72	2.87	0.80	7.48	3.02	0.78	7.25	3.19	0.80	7.27	2.93	0.78
	LatentAct-Diff	6.53	2.62	0.78	7.23	2.81	0.78	7.11	3.10	0.79	6.70	2.84	0.78
No hands	**Forecasting**												
	LatentAct	7.93	2.97	0.76	8.26	3.01	0.76	8.19	3.28	0.77	7.90	3.00	0.75
	LatentAct-Diff	7.82	2.89	0.71	8.63	3.06	0.71	8.19	3.15	0.72	7.93	2.96	0.73
	Interpolation												
	LatentAct	7.13	2.89	0.79	7.24	2.97	0.79	7.44	3.21	0.80	7.12	2.89	0.78
	LatentAct-Diff	6.60	2.65	0.76	7.73	2.78	0.76	7.42	3.04	0.78	6.89	2.71	0.77

Table 3. **LatentAct-Diff trends**. Training InterPred with diffusion loss leads to better hand poses but worse contact maps than LatentAct.

Method	MPJPE↓	MPJPE-PA↓	F1↑
HCTFormer	7.52	3.02	0.76
HCTFormer + InterCode	6.72	2.87	0.80
HCTDiff	8.30	2.72	0.77
HCTDiff + InterCode	6.53	2.62	0.78

Table 4. Two stage training with InterCode improves over single-stage training methods HCTFormer & HCTDiff. Note that HCT-Former + InterCode is same as LatentAct.

Method	MPJPE↓	MPJPE-PA↓	F1↑
HCTFormer	7.52	3.02	0.76
− no contact map	7.65	2.89	–
LatentAct	6.72	2.87	0.80
− no contact map	7.76	2.91	–
HCTDiff	8.30	2.72	0.77
− no contact map	8.43	2.78	–
LatentAct-Diff	6.53	2.62	0.78
− no contact map	7.09	2.90	–

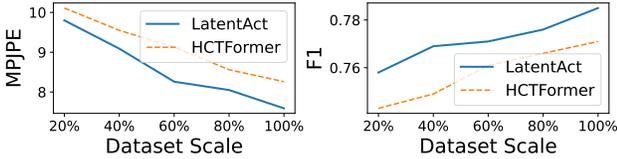Table 5. Loss on contact maps helps hand predictions for all models.

Figure 5. More training data helps both LatentAct & HCTFormer.

Dataset scale: In Fig. 5, we see that both LatentAct & HCTFormer improve with larger training dataset size.

4.3. Visualizations

We compare the predictions of LatentAct & the best baseline on a few examples from task level generalization setting (Fig. 6) for both forecasting & interpolation for 3 timesteps (t=5, 15, 25). The left column shows the input image with the contact point (projected in the image in cyan blob), goal image (for interpolation task) & text describing the action. The other columns show: (a) Camera View: predicted hand in the camera view with the contact point, this captures the placement of the hand around the contact point, (b) Another View: this better visualizes the hand articulation from a different camera viewpoint, (c) Contact Maps are shown as red parts of the hand mesh. LatentAct leads to better orientation of hand & sharper contact maps than the baseline. Note that these tasks are not seen during training. More visualizations & analysis are provided in the supplementary.
Limitations: We do not predict the object state change. While 3D object models are not available, alternatives like

predicting 2D object masks could be explored. While we assume a 3D contact point to be available, it can also be estimated using off-the-shelf depth models & ground truth intrinsics to get the 3D location of a 2D point that can be prompted by a user (*e.g.* by clicking a point on the image). Our experiments measure whether the predicted interaction trajectory is accurate, but it does not account for the multi-modal nature of future prediction. The annotations in our work are programatically generated and may not be accurate in all cases. While we filter out the erroneous cases, using a better annotation tool [47] would be beneficial.

5. Conclusion

Our model predicts interaction trajectories, *i.e.* 3D hand motion & contact maps, from single RGB view, action text & 3D contact point as input. It consists of: (1) Interaction Codebook: a VQVAE to learn a latent codebook of 3D hand poses & contact points, *i.e.*, tokenizing interactions. (2) Interaction Predictor: a transformer-decoder module to predict the interaction trajectory from test time inputs, by using an indexer module to retrieve a latent interaction from the learned codebook. Our large-scale experiments on the diverse HoloAssist dataset show benefits across 4 generalization settings. Our work is a first step towards learning a motion prior on interaction sequences and can be augmented
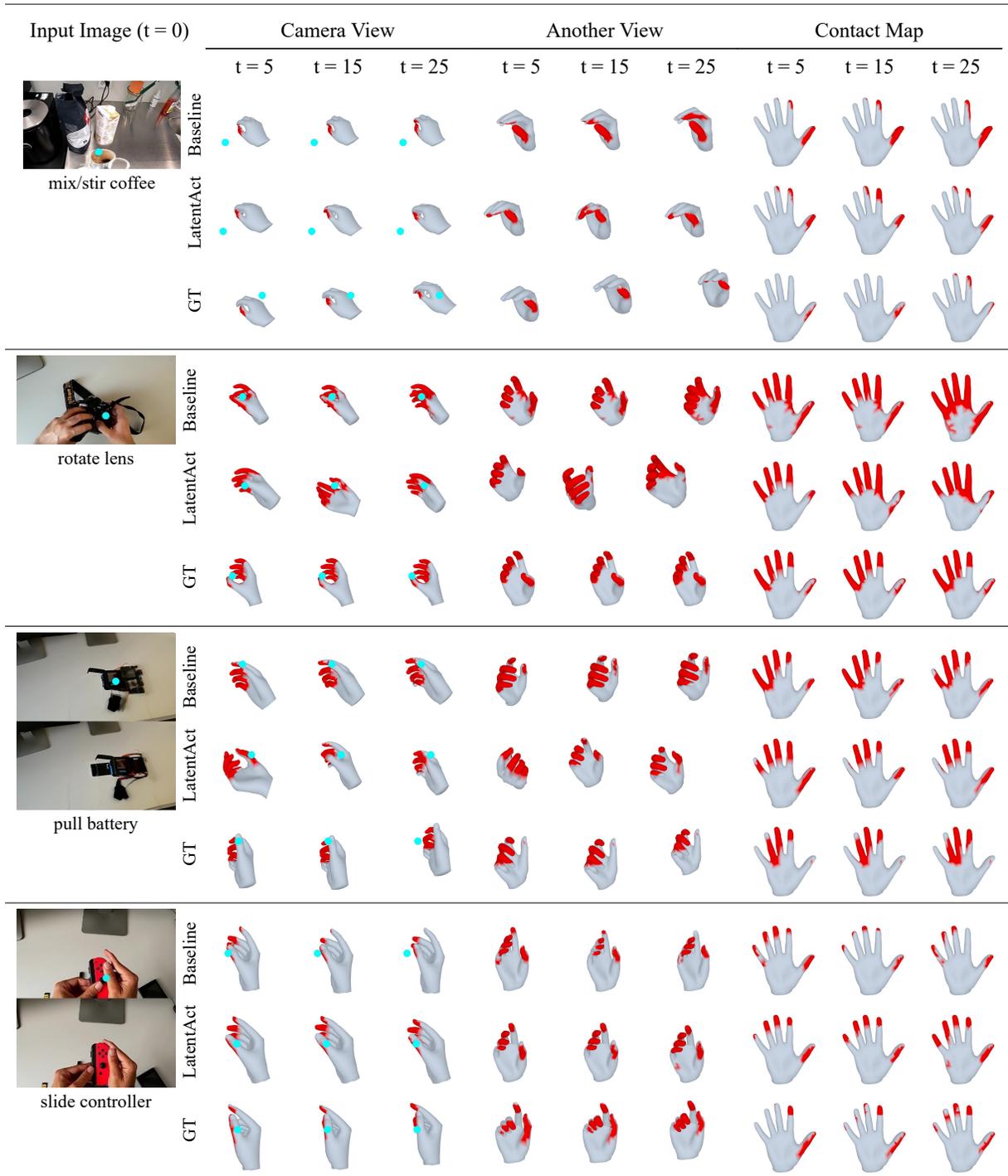
Figure 6. **Visualizations**. We compare the predictions of LatentAct & the best baseline on a few examples from the task level generalization setting for both forecasting & interpolation for 3 timesteps (t=5, 15, 25). The left column shows the input image with the contact point (projected in the image in cyan blob), goal image (for interpolation task) & text describing the action. The other columns show: (a) Camera View: predicted hand in the camera view with the contact point, this captures the placement of the hand around the contact point, (b) Another View: this better visualizes the hand pose from a different camera viewpoint, (c) Contact Maps are shown as red parts of the hand mesh. LatentAct leads to better orientation of hand & sharper contact maps than the baseline. *Note that these tasks are not seen during training.*

with object trajectories with better 3D models in the future. Our data pipeline is modular and can also incorporate better

hand & segmentation models in future.

References

[1] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[2] Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Fan Zhang, Jade Fountain, Edward Miller, Selen Basol, Richard Newcombe, Robert Wang, Jakob Julian Engel, and Tomas Hodan. Introducing hot3d: An egocentric dataset for 3d hand and object tracking. *arXiv*, 2406.09598, 2024. 2, 5

[3] Jona Braun, Sammy Joe Christen, Muhammed Kocabas, Emre Aksan, and Otmar Hilliges. Physically plausible full-body hand-object interaction synthesis. *arXiv*, 2309.07907, 2023. 2

[4] Junuk Cha, Jihyeon Kim, Jae Shin Yoon, and Seungryul Baek. Text2hoi: Text-guided 3d motion generation for hand-object interaction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 5

[5] Matthew Chang, Aditya Prakash, and Saurabh Gupta. Look ma, no hands! agent-environment factorization of egocentric videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 3

[6] Tianyi Cheng, Dandan Shan, Ayda Sultan Hassen, Richard Ely Locke Higgins, and David Fouhey. Towards a richer 2d understanding of hands at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 2, 5

[7] Sammy Christen, Shreyas Hampali, Fadime Sener, Edoardo Remelli, Tomas Hodan, Eric Sauser, Shugao Ma, and Bugra Tekin. Diffh2o: Diffusion-based synthesis of hand-object interactions from textual descriptions. *arXiv*, 2403.17827, 2024. 2, 3, 5

[8] 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)*, 2018. 2

[9] 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 *NeurIPS Track on Datasets and Benchmarks*, 2022. 2

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 5

[11] Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J. Black, and Otmar Hilliges. ARCTIC: A dataset for dexterous bimanual hand-object manipulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 3, 5, 6

[12] Zicong Fan, Takehiko Ohkawa, Linlin Yang, Nie Lin, Zhishan Zhou, Shihao Zhou, Jiajun Liang, Zhong Gao, Xuanyang Zhang, Xue Zhang, Fei Li, Zheng Liu, Feng Lu, Karim Abou Zeid, Bastian Leibe, Jeongwan On, Seungryul Baek, Aditya Prakash, Saurabh Gupta, Kun He, Yoichi Sato, Otmar Hilliges, Hyung Jin Chang, and Angela Yao. Benchmarks and challenges in pose estimation for egocentric hand interactions with objects. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 3

[13] Anindita Ghosh, Rishabh Dabral, Vladislav Golyanik, Christian Theobalt, and Philipp Slusallek. Imos: Intent-driven full-body motion synthesis for human-object interactions. *EuroGraphics*, 42(2):1–12, 2023. 2

[14] J.J. Gibson. The ecological approach to visual perception, 1979. 3

[15] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[16] Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3

[17] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[18] Shreyas Hampali, Sayan Deb Sarkar, Mahdi Rad, and Vincent Lepetit. Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[19] Mohamed Hassan, Partha Ghosh, Joachim Tesch, Dimitrios Tzionas, and Michael J. Black. Populating 3d scenes by learning human-scene interaction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[20] Di Huang, Xiaopeng Ji, Xingyi He, Jiaming Sun, Tong He, Qing Shuai, Wanli Ouyang, and Xiaowei Zhou. Reconstructing hand-held objects from monocular video. In *ACM Transactions on Graphics*, 2022. 2

[21] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. 3

[22] Jaewoo Jeong, Daehee Park, and Kuk-Jin Yoon. Multi-agent long-term 3d human pose forecasting via interaction-aware

trajectory conditioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 5, 6

[23] Hanwen Jiang, Shaowei Liu, Jiashun Wang, and Xiaolong Wang. Hand-object contact consistency reasoning for human grasps generation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 3

[24] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2o: Two hands manipulating objects for first person interaction recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[25] Gen Li, Varun Jampani, Deqing Sun, and Laura Sevilla-Lara. LOCATE: localize and transfer object parts for weakly supervised affordance grounding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[26] Quanzhou Li, Jingbo Wang, Chen Change Loy, and Bo Dai. Task-oriented human-object interactions generation with implicit neural representations. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2024. 2

[27] Shaowei Liu, Yang Zhou, Jimei Yang, Saurabh Gupta, and Shenlong Wang. Contactgen: Generative contact modeling for grasp generation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023. 3

[28] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 5

[29] Takehiko Ohkawa, Kun He, Fadime Sener, Tomas Hodan, Luan Tran, and Cem Keskin. Assemblyhands: Towards egocentric activity understanding via 3d hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12999–13008, 2023. 2

[30] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3d with transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 5

[31] Aditya Prakash, Matthew Chang, Matthew Jin, and Saurabh Gupta. 3d reconstruction of objects in hands without real world 3d supervision. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 2

[32] Aditya Prakash, Arjun Gupta, and Saurabh Gupta. Mitigating perspective distortion-induced shape ambiguity in image crops. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 3

[33] Aditya Prakash, Ruisen Tu, Matthew Chang, and Saurabh Gupta. 3d hand pose estimation in everyday egocentric images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 2

[34] Sergey Prokudin, Christoph Lassner, and Javier Romero. Efficient learning on point clouds with basis point sets. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021. 3, 4

[36] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv*, 2408.00714, 2024. 2, 5

[37] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6):245:1–245:17, 2017. 2, 3

[38] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (ToG)*, 2017. 3, 5

[39] Rómer Rosales and Stan Sclaroff. Combining generative and discriminative models in a framework for articulated pose estimation. *International Journal of Computer Vision*, 67:251–276, 2006. 2

[40] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F Fouhey. Understanding human hands in contact at internet scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[41] Rajinder S Sodhi, Brett R Jones, David Forsyth, Brian P Bailey, and Giuliano Maciocci. Bethere: 3d mobile collaboration with spatial input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 179–188, 2013. 2

[42] Mohan Kumar Srirama, Sudeep Dasari, Shikhar Bahl, and Abhinav Gupta. HRP: human affordances for robotic pretraining. *arxiv*, 2407.18911, 2024. 3

[43] Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2, 3, 5

[44] Omid Taheri, Yi Zhou, Dimitrios Tzionas, Yang Zhou, Duygu Ceylan, Soren Pirk, and Michael J. Black. GRIP: Generating interaction poses using latent consistency and spatial cues. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2024. 2, 3, 5

[45] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit Haim Bermano. Human motion diffusion model. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. 5, 6

[46] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021. 3, 4

[47] Shashank Tripathi, Agniv Chatterjee, Jean-Claude Passy, Hongwei Yi, Dimitrios Tzionas, and Michael J. Black. DECO:

dense estimation of 3d human-scene contact in the wild. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023. 3, 7

[48] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2

[49] Sirui Xu, Zhengyuan Li, Yu-Xiong Wang, and Liang-Yan Gui. Interdiff: Generating 3d human-object interactions with physics-informed diffusion. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023. 3

[50] Yuhang Yang, Wei Zhai, Hongchen Luo, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Grounding 3d object affordance from 2d interactions in images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023. 3

[51] Yufei Ye, Abhinav Gupta, and Shubham Tulsiani. What's in your hands? 3D reconstruction of generic objects in hands. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[52] Yufei Ye, Poorvi Hebbar, Abhinav Gupta, and Shubham Tulsiani. Diffusion-guided reconstruction of everyday hand-object interaction clips. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023. 5

[53] Yufei Ye, Xueting Li, Abhinav Gupta, Shalini De Mello, Stan Birchfield, Jiaming Song, Shubham Tulsiani, and Sifei Liu. Affordance diffusion: Synthesizing hand-object interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 3

[54] Yufei Ye, Abhinav Gupta, Kris Kitani, and Shubham Tulsiani. G-hop: Generative hand-object prior for interaction reconstruction and grasp synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3, 5

[55] Hui Zhang, Sammy Joe Christen, Zicong Fan, Luocheng Zheng, Jemin Hwangbo, Jie Song, and Otmar Hilliges. Artigrasp: Physically plausible synthesis of bi-manual dexterous grasping and articulation. *arXiv*, 2309.03891, 2023. 2