

# GRIP: Generating Interaction Poses Using Spatial Cues and Latent Consistency

Omid Taheri<sup>1</sup> Yi Zhou<sup>2</sup> Dimitrios Tzionas<sup>3</sup> Yang Zhou<sup>2</sup>  
Duygu Ceylan<sup>2</sup> Soren Pirk<sup>4</sup> Michael J. Black<sup>1</sup>

<sup>1</sup>Max Planck Institute for Intelligent Systems, Germany <sup>2</sup>Adobe Research <sup>3</sup>University of Amsterdam <sup>4</sup>Kiel University

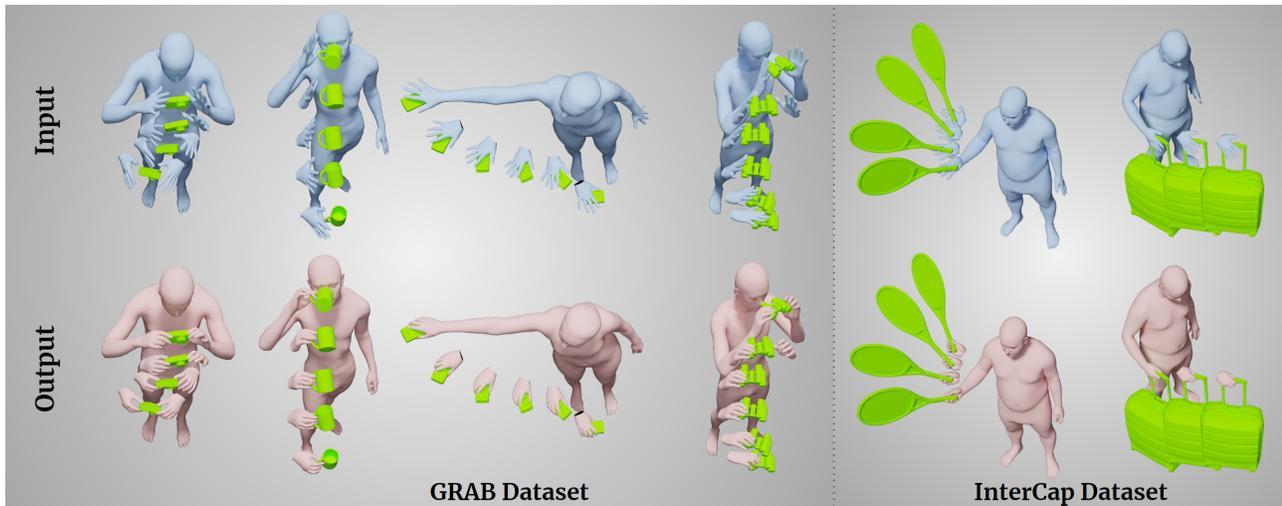


Figure 1. GRIP generates realistic hand-object interaction poses (pink), given the easy-to-acquire body and object motion without fingers (blue) – notice that the input hand pose is constant. GRIP animates the hands to be consistent with the body and object, producing realistic poses in various scenarios like pre-/post-grasp hand opening, and single or bi-manual grasps. It also works with various object shapes and sizes, and on different datasets like GRAB [51] (left) and InterCap [25] (right).

## Abstract

*Hands are dexterous and highly versatile manipulators that are central to how humans interact with objects and their environment. Consequently, modeling realistic hand-object interactions, including the subtle motion of individual fingers, is critical for applications in computer graphics, computer vision, and mixed reality. Prior work on capturing and modeling humans interacting with objects in 3D focuses on the body and object motion, often ignoring hand pose. In contrast, we introduce GRIP, a learning-based method that takes, as input, the 3D motion of the body and the object, and synthesizes realistic motion for both hands before, during, and after object interaction. As a preliminary step before synthesizing the hand motion, we first use a network, ANet, to denoise the arm motion. Then, we leverage the spatio-temporal relationship between the body and the object to extract novel temporal interaction cues, and use them in a two-stage inference pipeline to generate the hand motion. In the first stage, we introduce a new approach to encourage motion temporal consistency*

*in the latent space (LTC) and generate consistent interaction motions. In the second stage, GRIP generates refined hand poses to avoid hand-object penetrations. Given sequences of noisy body and object motion, GRIP “upgrades” them to include hand-object interaction. Quantitative experiments and perceptual studies demonstrate that GRIP outperforms baseline methods and generalizes to unseen objects and motions from different motion-capture datasets. Our models and code are available for research purposes at <https://grip.is.tue.mpg.de>.*

## 1. Introduction

Digital humans that move and interact naturally with 3D worlds have many applications in data creation, games, XR, and telepresence. In particular, physically-plausible hand-object interaction is critical for realism. Unfortunately, automatically generating hand motions consistent with the world is challenging and no fully-general solutions exist.

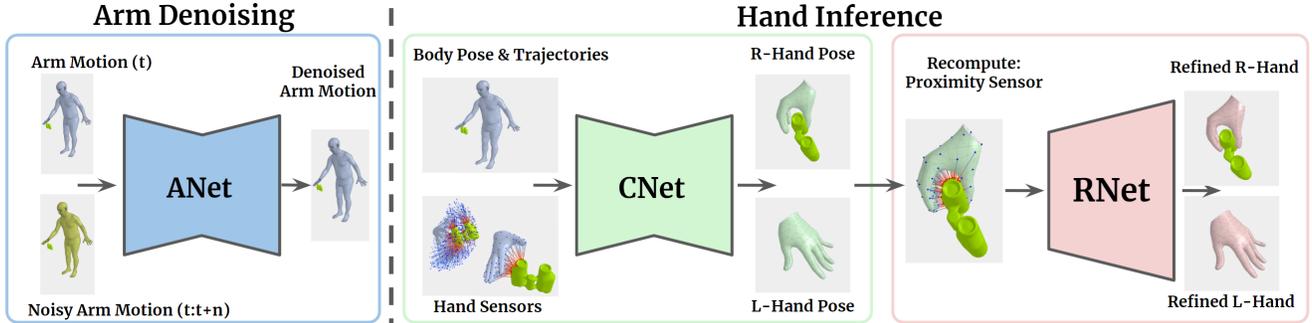


Figure 2. Overview of GRIP. We first denoise the arm motion using the ANet network. We then predict hand interaction motion in two stages: (CNet) Given the hand-object spatial features, extracted using our *hand sensors*, body pose and trajectories in two consecutive frames, CNet predicts both left- and right-hand poses. (RNet) Based on the predicted hand poses, we recompute the *proximity sensor* feature and refine the hand poses with RNet to enhance interaction accuracy and reduce possible penetrations.

The problem is challenging since different object shapes require different types of interaction and hand grasps, such as a power grasp of an apple, a delicate three-finger pinching of a cup handle, and bi-manual grasp of binoculars. Performing these actions is effortless for humans; however, even small errors, such as hand-object penetrations or subtly misplaced arms or fingers, can significantly affect the perceived realism of generated grasps for virtual avatars.

Here we consider generating realistic grasps where a 3D animation of the body and object is given, either from motion capture (MoCap), reconstructed from videos, or from an animator. Motion capture data rarely contains hands as they are difficult to capture, requiring small markers that are often occluded and require high-resolution cameras. In some cases, despite being tracked, hands and arms are noisy [25]. Objects, in contrast, are easier to track. Figure 1-top illustrates this scenario with body and object motion, from GRAB [51] and InterCap [25], but only rigid hands. The goal is to transform this data into a more natural animation by synthesizing the appropriate hand-object interaction, as illustrated in Fig. 1-bottom. With this approach we can “upgrade” existing datasets to support research on human-object interaction.

To this end, we introduce *GRIP*, which stands for *Generating Realistic Interaction Poses*, a learned model that generates realistic hand motions for interactions with various unseen objects. Previous work focuses only on static grasping [28, 51], requires an initial hand pose that is then improved [18, 61], or only considers single-hand grasps and mirrors them for the other hand [58, 61]. Going beyond these approaches, our method directly infers dynamic hand motion, both in a single-hand or bimanual scenario, conditioned only on the object and body motion.

Our contributions are two-fold. First, we propose a set of virtual “hand sensors” to extract rich spatio-temporal interaction cues between the body and the object. Specifically, we introduce an *Ambient Sensor* that senses the object shape and motion within the hands’ broader reaching region, as

well as a *Proximity Sensor* that captures fine-grained geometric features and a more nuanced distance field between the hand and object surface within the hands’ closer region. While virtual sensors have been used in prior work, our novel contribution is the innovative use of a distance-based representation combined with interaction-aware attention [52]. This unique combination significantly improves results and generalizes to unseen objects and motions.

Second, we propose an *arm denoising* network and a novel two-stage *hand inference* pipeline to leverage these features and generate realistic interaction motions. Since arm motions from tracking or reconstruction can be noisy, we first use an arm denoising network, ANet to refine arm motion. Our goal for hand inference is to achieve near real-time performance. For this, unlike iterative optimization in previous methods, we employ two networks to generate course and refined motions. First, the *Consistency Network (CNet)* takes features from both *Hand Sensors* and generates smooth and consistent hand interaction motions. Achieving this is challenging, as motions need to be realistic, temporally consistent, and natural. Naively applying temporal smoothness terms to the final output hand motion, cf. [50, 52], breaks the contact consistency and leads to high-frequency changes in contact areas. To overcome this, we propose a novel *Latent Temporal Consistency (LTC)* solution. Specifically, we jointly learn global and residual latent codes to represent two successive frames and apply temporal consistency in a latent space, as shown in Fig. 4. Then, to mitigate any inconsistency between the two global latent codes, the key insight is to decode them using a “shared” network to generate consistent hand poses. We use LTC in both ANet and CNet to ensure consistency in the motions.

The generated hand poses from CNet bring fingers very close to the object surface, allowing the *Proximity Sensor* to capture a more nuanced distance field. Therefore, in the second stage, we recompute the *Proximity Sensor* features and use a refinement network, RNet, to add subtle refine-

ments and resolve penetrations in the interaction frames.

GRIP is trained to generate both left- and right-hand motion simultaneously, enabling realistic modeling of single-hand and bi-manual interactions. In contrast to other methods, which only focus on contact frames [18, 51], our model is able to generate dynamic hand motions *before*, *during*, and *after* the interaction with objects. Additionally, unlike work [61] that requires expensive optimization in the pose refinement step, our framework consists only of feed-forward neural networks. By predicting realistic hand and finger motions, GRIP can be used to increase the realism of an avatar’s interaction in AR/VR applications, refine noisy hand-object interaction motions (Fig. 6-left), enrich existing interaction datasets that do not contain realistic finger motions (Fig. 6-right), or capture new datasets with dexterous interactions but without explicitly tracking fingers.

We evaluate GRIP quantitatively and qualitatively on a withheld test set from the GRAB dataset, with 5 unseen objects and motions. The results show that our method generates accurate hand motions involving object grasping and manipulation. We also show that GRIP generalizes to other MoCap datasets and larger objects, not present in GRAB, by generating hand grasps for unseen objects from the MoGaze [30] and InterCap [25] datasets (see Fig. 6). The quantitative evaluation shows that GRIP outperforms baselines, while our ablation studies explore the efficacy of our *latent temporal consistency*, *hand sensors*, and other design choices. Finally, we perform a perceptual study to evaluate the quality of the generated hand interaction motions. The results indicate that hand-object interaction sequences generated by GRIP achieve a level of realism similar to GRAB’s ground-truth motions.

## 2. Related Work

Despite the many advances in the field of motion synthesis for human avatars, generating accurate hand motion is still a challenging and unsolved problem. While many approaches focus on improving static grasps [18, 51] with manually designed heuristics [22, 23], more recent techniques consider dynamic grasp generation [58, 61]. Such methods are still limited, and we review the most relevant ones below.

**Static Grasp Generation:** Generating static grasps has been widely studied in robotics, computer graphics, and computer vision. Common approaches in graphics and robotics use physics-based control to generate novel hand grasps for a given 3D object. This includes using reference poses to optimize generated grasps [47], using hand pose and force closure [13, 31], or pruning grasp candidates through physics-based analysis [5, 34, 35, 39]. Some recent methods take a data-driven approach and generate hand grasps by training on large hand-object interaction datasets [6, 7, 11, 26, 28, 29, 51, 63]. Most of these approaches either estimate the grasping-hand pose directly [6, 7, 26],

based on model parameters [37, 41] or by employing an implicit representation [28, 61]. Other approaches further refine the initially generated grasps by using a neural network [51] or by leveraging predicted contact maps [18, 26].

**Dynamic Grasp Generation:** Generating hand-object grasping motions is more challenging than static grasp generation. Most previous methods approach this by generating contact constraints and by resolving them through optimization-based methods [36, 39, 57, 60]. Despite being physically plausible, the generated hand motions lack realism and are prone to interaction artifacts. More recently, reinforcement learning (RL) has been used for hand-only and full-body scenarios [2, 3, 40, 43, 44, 48, 49]. Christen et al. [10] employ physics simulation along with RL for dynamic grasp synthesis; however, their method requires reference hand-grasps and dynamic features of the object. A key challenge of these methods is generalization to new object geometries and hand configurations. Zhang et al. [58] use a distance-based spatial representation between hands and objects and train a network to generate right-handed object manipulation motions. To avoid interaction artifacts, Zhou et al. [61] propose an object-centric spatio-temporal representation and refine it with a neural network. The refined representation is used in an optimization to recover the hand-interaction motion. Unlike our approach, most of these methods treat each hand separately, making generated hand-collaboration and bi-manual grasps unrealistic.

**Object and Scene Interaction:** Some early work uses foot and hand contact annotations from MoCap datasets with optimization-based methods to extend or retarget human motions to scenes [17, 27, 32, 33]. Alternatively, deep reinforcement learning can be used to generate body-scene [8, 42, 44] or hand-object [9, 10, 16] interactions. Other methods use descriptors for dynamic interactions [45, 46], encode the joint motions of humans w.r.t. scene points [1], or use Laplacian deformation between body and object vertices to define a representation for modeling interactions [24]. As geometry-based approaches are not robust to real-world noise, some methods take a data-driven approach to predict action and motion sequences [55] or to generate key frames of motions and then complete them with data-driven or optimization-based techniques [20, 52, 56].

**Hand-Object Interaction Tracking:** For graphics applications, hand motions have traditionally been animated by artists [58]. While MoCap can be used to capture hand motion datasets [6, 7, 15, 19, 54], such captures are technically challenging, limiting the amount of such data in the world. For the MoGaze [30], KIT [38], and BEHAVE [4] datasets, human motions are tracked during interaction with objects, but the fingers and palm, are not explicitly captured. Taheri et al. [51] and Fan et al. [14] capture accurate hand-object interactions with a high-end MoCap system, but this approach does not scale. Zhang et al. [58] propose a method

for real-time hand motion synthesis, given the wrist and object motion. However, this barely generalizes to new object shapes and full-body motions. InterCap [25] captures whole-body interactions with objects, but hands are noisy.

**Summary:** Previous methods suffer from one or more of generalization ability, computation time, an initial hand pose requirement, or model only single-hand interactions. Our data-driven method, GRIP, addresses these limitations and efficiently generates realistic motions for both hands interacting with novel objects.

### 3. Method

Our goal is to add realistic hand poses to a body, based on the relative motion of the body and object during an interaction. To correctly estimate the hand interaction motion, we need to model how and when the object grasp happens. These cues can be found in the object’s geometry and the correlated body-object motion trajectories. For example, if the distance between a wrist and the object is decreasing, the hand is approaching the object, but if it becomes constant and the object starts moving, we can infer it is grasped.

To represent such information, we design two virtual “hand sensors”; (1) the *Ambient Sensor* obtains the object’s geometric features and its spatial relation to the hands and (2) the *Proximity Sensor* obtains a fine-grained distance field from different hand regions to the object surface.

However, if the arm motion is noisy, these computed features will also be inaccurate. Therefore, as a preliminary step, we use an arm denoising network, ANet, as shown in Fig. 2, which takes the noisy arm motion and refines it while enforcing the temporal motion consistency.

Then, we propose a two-stage hand prediction framework to generate hand motion, as illustrated in Fig. 2. In the first stage, since we do not have an initial hand pose, we use a mean hand to compute the features of the hand sensors to predict both hand poses. To consider temporal information, we feed our model with the body poses and the hand sensors’ features of the current and next frame, in addition to the hand-to-object distance and velocity in the next  $n$  frames (typically 10, but this can be varied). In the second stage, based on the predicted hand poses, we recompute the *Proximity Sensor* feature and refine the predictions to enhance interaction accuracy and reduce hand-object penetrations. Details about each hand sensor and the neural networks are provided below.

#### 3.1. Body and Hand Representations

To model the body and hand motion, we use the SMPL-X [41] model. It can represent fine-detailed motion and accurate physical interactions, which are critical for object-interaction motions. Based on the body shape,  $\beta$ , and pose,  $\theta$ , parameters, SMPL-X reconstructs the body surface using linear blend skinning with a learned rigged

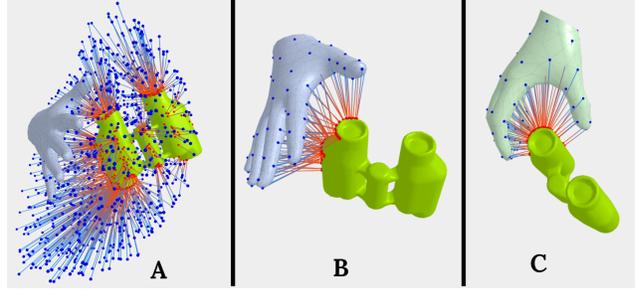


Figure 3. Visualization of our Hand Sensors (only right-hand for simplicity). (A) *Ambient Sensor* points (blue) and their computed distances to the closest object points (red). This captures the object geometry and distance to the hands. (B) *Proximity Sensor* feature computation for CNet’s inputs with mean-hand pose initialization. (C) Recomputing the *Proximity Sensor* values for RNet, using the hand poses generated by CNet. Note that the corresponding points on the object change for each finger compared to (B).

skeleton,  $\mathcal{J} \in \mathbb{R}^{55 \times 3}$ . The full set of SMPL-X parameters,  $\Theta = \{\theta \in \mathbb{R}^{55 \times 6}, \gamma \in \mathbb{R}^3\}$  includes both hands. Here, we predict only the parameters of the hands: the right-hand pose,  $\theta^r \in \mathbb{R}^{15 \times 6}$  and the left-hand pose,  $\theta^l \in \mathbb{R}^{15 \times 6}$  [62]. In addition, to efficiently represent the hand surface, we follow [52] and sample 99 vertices on each hand; these are denoted  $v^l$  and  $v^r$ , for the left and right hand, respectively.

#### 3.2. Ambient Sensor

To sense the location and shape of the object, we uniformly sample a set of 1024 points in a hemisphere that is rigidly attached to each hand and centered at the base middle-finger joint, as shown in Fig. 3-A. For each motion frame, we compute the distance,  $d$ , from each of these points to the closest vertex on the object surface. This allows us to capture detailed information about the object shape and the relative distance between the hands and the object. The former informs the hand pose to adapt to certain shapes, while the latter helps predict the state of the hand motion, such as the pre-grasp and pre-release opening, and to keep consistent contact during the interaction.

Unlike commonly used voxel grids [50, 58], which provide a binary and discrete spatial representation, our novel *Ambient Sensor* provides a continuous representation as it uses a distance-based representation. Furthermore, we pass the distances,  $d$ , through the interaction-aware attention transformation (Eq. (1)) proposed by [52], with  $w = 5$ , to emphasize points closer to the object surface

$$I_w(d) = \exp(-w \times d), \quad w > 0. \quad (1)$$

The ablation studies in Tab. 2 and comparison with voxel-based ambient sensors show these unique combination captures rich spatial hand-object relations, improves results, and generalizes to unseen objects and motions.

### 3.3. Proximity Sensor

Although the *Ambient Sensors* capture important interaction information, they do not encode the distance of specific hand regions to the surface of the object; this is essential to know the contact areas. Therefore, we use the sampled hand vertices  $v$  and compute their closest distance to the object surface. Since we do not have the hand pose in the beginning, we initialize the hand with SMPL-X’s [41] mean hand pose and compute the proximity features in the first stage of prediction, as shown in Fig. 3-B. In the second stage, we recompute *Proximity Sensors*’ values using the hand poses generated from the first stage, as shown in Fig. 3-C.

In contrast to the *Ambient Sensor*, the *Proximity Sensor* provides fine-grained geometric details. This more nuanced information about interaction is essential to generate hand poses with fewer penetrations and better contacts, particularly when the hands are very close to the object’s surface. Thus, for the *Proximity Sensor*, we apply the transformation in Eq. (1) with a higher weight ( $w = 50$ ) w.r.t. the *Ambient Sensor*, to put emphasis on the vertices closer to the object.

### 3.4. Consistency Network (CNet)

CNet is a novel encoder-decoder neural network that takes the body motion and hand sensor features of two consecutive frames at time  $t$  and  $t + 1$  to predict the hand poses of both frames. The two frames will be used in our proposed *Latent Temporal Consistency (LTC)* algorithm to enforce temporal and contact consistency for the final prediction. CNet additionally takes the average hand-to-object distance  $d$  in the future  $n$  frames, from  $t$  to  $t + n$ , where  $n = 10$  by default, as input to better disambiguate the grasp and release moments. The detailed architecture of CNet is illustrated in Fig. 4. The inputs to the network are:

$$X = \left[ \beta, \theta_{t:t+1}, h_{t:t+1}^A, h_{t:t+1}^P, \bar{d}_{t:t+n}, \bar{\dot{d}}_{t:t+n} \right] \quad (2)$$

where  $t : t + i$  denotes  $i$  motion frames in the future including the current frame,  $\theta_{t:t+1}$  are the SMPL-X joint angles without considering the global root joint,  $h_{t:t+1}^A$  and  $h_{t:t+1}^P$  are the hand *Ambient Sensor* and *Proximity Sensor* values for both left and right hands, and  $\bar{d}_{t:t+n}$  and  $\bar{\dot{d}}_{t:t+n}$  are the average of hand-to-object distance and its rate of change for sampled hand vertices in the  $n$  future frames.

**Latent Temporal Consistency (LTC):** In addition to physically plausible hand-object contact, an important factor in the realism of interaction motions is consistent dynamics and contact areas between consecutive frames. To enforce these, we smooth the motion in the latent space of hand motions rather than in the output space, as we noticed the latter adds high-frequency noise to the contact areas throughout the motion. As shown in Fig. 4, the encoder,  $\mathcal{E}^C$ , maps the input  $X$  to two latent codes,  $z_t, z_{t+1}^t \in \mathbb{R}^{256}$ , where  $z_t$  denotes the global latent code for a hand pose in

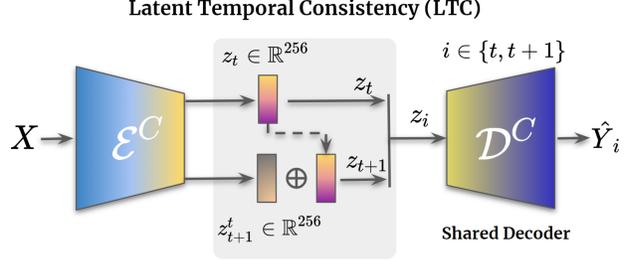


Figure 4. CNet Architecture. We propose the LTC algorithm that enforces consistency between two successive frames in the latent space (see Sec. 3.4 for more details).

the current frame and  $z_{t+1}^t$  is the relative latent code for the next frame with respect to the current frame. We compute the global latent code for the next frame by adding the two latent codes as  $z_{t+1} = z_t + z_{t+1}^t$ ; see Fig. 4. We then pass each global latent code individually to a shared decoder,  $\mathcal{D}^C$ , to get the outputs  $\hat{Y}$ , for the current and next frame. The shared decoder helps regulate inconsistency between the two global latent codes, as it is represented and penalized in the final generated hand poses. The output of CNet is:

$$\hat{Y} = \left[ \hat{\theta}_{t:t+1}^r, \hat{\theta}_{t:t+1}^l, \hat{h}_{t:t+1}^P \right] \quad (3)$$

where  $\hat{\theta}_{t:t+1}^r, \hat{\theta}_{t:t+1}^l \in \mathbb{R}^{15 \times 6}$  are right-/left-hand poses in the current and next frame, and  $\hat{h}_{t:t+1}^P$  are the inferred *Proximity Sensor* values; the latter ones have been shown to increase realism and lower errors [52].

Generating hand poses in the current and next frame allows for defining consistency and smoothness losses between them. Evaluations in Tab. 2 show that the motions generated with our LTC algorithm achieve a lower error and better consistency compared to baselines with no enforced consistency or with consistency in the output space.

We use fully-connected dense residual blocks with skip connections for both the encoder and decoder, and train CNet end-to-end. The training loss is defined as

$$\mathcal{L} = \lambda_v \mathcal{L}_v + \lambda_{h^P} \mathcal{L}_{h^P} + \lambda_\theta \mathcal{L}_\theta, \quad (4)$$

where  $\mathcal{L}_v = \|v - \hat{v}\|_1$  is a loss on the hand vertices  $v$ ,  $\mathcal{L}_\theta = \|\hat{\theta}^l - \theta^l\|_2 + \|\hat{\theta}^r - \theta^r\|_2$  is on the joint rotations of both hands and  $\mathcal{L}_{h^P} = \|\hat{h}^P - h^P\|_1$  is on the hand-to-object distances, both directly estimated from the network and derived from the estimated hand poses.

### 3.5. Arm Denoising Network (ANet)

For the hand sensors in CNet to capture rich information between the hand and the object, the motion of these two should be very accurate and without noise. Therefore, as shown in Fig. 2-left, we train ANet to first refine the arm

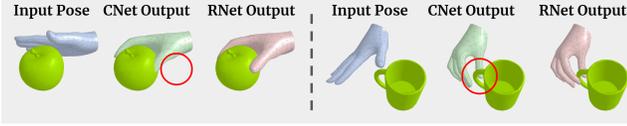


Figure 5. Comparing CNet and RNet generated grasps. Results show that RNet effectively refines the penetration and “non-contact” artifacts (red circles) of the CNet results.

motion before passing to CNet. It takes as input both arms’ pose in the current frame,  $\theta^{la}$  and  $\theta^{ra}$ , and the noisy poses of the future frame,  $\theta_p^{la}$  and  $\theta_p^{ra}$ , along with the hand sensor features, and gives the denoised arm poses. We use a similar architecture to CNet, and enforce the consistency between the denoised poses in the latent space of the network using LTC. For more details about ANet please see *Sup. Mat.*

### 3.6. Refinement Network (RNet)

The motions generated by CNet are in the right ballpark but can be refined further to improve realism and remove possible penetrations. To this end, we train a refinement network, RNet. We use the generated hand poses from CNet to recompute *Proximity Sensor* features,  $h_\theta^P$ , similar to CNet inputs (see Fig. 3-C). Then RNet takes  $h_\theta^P$  and the hand poses,  $\theta^l$  and  $\theta^r$ , and outputs the refined hand poses. To keep the motion dynamics, generated from CNet, we train RNet to refine hand poses only in the interaction frames and not to change the pose when hands are far away from the object surface. In addition to the CNet output, we train RNet on perturbed training data to simulate noisy inputs. Training losses are similar to those used for CNet in Eq. (4). RNet consists of 3 fully-connected residual layers with skip connections in between, for an architectural overview, more details, and the data processing pipeline please see *Sup. Mat.*

## 4. Experiments

### 4.1. Evaluation Metrics

We use the standard “Mean Per-Joint Position Error” (MPJPE) and “Mean Per-Vertex Position Error” (MPVPE), which represent the Euclidean distance between the ground-truth and estimated hand joints and vertices, respectively.

**Intersection Volume (IV):** This measures the intersection volume between the hand and the object to assess the realism, i.e., the physical plausibility, of the generated grasps.

**Contact Consistency (CC):** This evaluates the consistency of contacts for the grasping frames of generated grasp motions, i.e., the finger sliding on the object surface. We use ground-truth motions to select grasp frames, and, for generated motions, compute the deviation distance from the contact areas on the object; for more details see *Sup. Mat.*

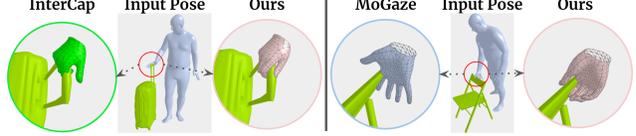


Figure 6. Our generated grasps (pink circles) for large objects from InterCap [25] and MoGaze [30], and comparison with the original grasps from these datasets.

### 4.2. Qualitative Evaluation

Results show that CNet generates reasonable and smooth hand grasps, but sometimes with artifacts like hand-object interpenetration. After applying the refinement network, RNet, the results look more realistic and physically plausible. In Fig. 5 we show examples of generated grasps using CNet and after applying the RNet refinement.

Figures 1 and 7 show several representative hand motions generated with GRIP, including pre-/post-grasp hand opening, single-hand grasps, and bi-manual grasps for different unseen object shapes. Overall, the generated hand motions are reasonable, smooth, and consistent. For more results, please see the accompanying *video* and *Sup. Mat.*

**Performance on Other Datasets:** GRIP is trained on the GRAB dataset, which only has small hand-held objects. High-quality data of hand-object interaction with large objects is rare. Despite training on small objects, our virtual hand sensors help generalize to larger objects, as they only sense the interaction areas locally and not the whole object. To highlight GRIP’s generalization capability, we show generated interaction poses for *unseen* large objects from the InterCap [25] and MoGaze [30] datasets in Fig. 6 and Fig. 1-right, and compare them with the original hand poses. For more results, see *Sup. Mat.*

**Cross-Object Grasp Transfer:** We show that GRIP can be used to transfer grasping motions from one object to another one, for the details and results please see *Sup. Mat.*

### 4.3. Ablation Study

**Latent Temporal Consistency (LTC):** To evaluate the importance of our proposed temporal consistency algorithm for interaction motions, we compare our network with two baselines, namely: (1) a network without enforced consistency (*w/o Consist.*) and (2) a network with consistency applied directly on the generated hand poses (*output Consist.*). As seen in Tab. 2-bottom our LTC method that smooths the latent space representation not only reduces the CC error, but also results in lower errors in MPVPE and MPJPE.

**Hand Sensors:** To evaluate the effect of our *Ambient Sensor* and *Proximity Sensor*, we train different baselines of GRIP by removing these features, (*w/o Ambient*) and (*w/o Proximity*), and additionally compare them to Voxel-based



Figure 7. GRIP results. We show various generated grasps, in single and bimanual scenarios, for different objects shapes. The input (flat, non-articulated) hands are shown with blue meshes, and GRIP’s generated hands (articulated) with pink meshes.

representation. We compare MPVPE, MPJPE, and CC between the generated hand motions and the ground truth. Results in Tab. 2-top show that our distance-based hand sensors provide rich interaction information to the network that leads to lower errors and consistent motions.

**RNet:** In Tab. 2 we evaluate our refinement network, RNet, by comparing the results of GRIP with RNet (*fullmodel*) and without it (*w/o RNet*). The table verifies that the refinement step helps reduce the hand MPJPE and MPVPE errors and enhance motion consistency.

**Number of Future Frames:** In Tab. 4-right we compare different variants of GRIP to show the effect of using a different number of future motion frames on the accuracy of the generated hand poses. The table verifies using more future frames (up to 10 frames) lets the network generate more accurate poses. This is a trade-off between a real-time performance (row 1) and a higher accuracy with some latency (rows 2-4). Empirically, we observe that performance saturates for more than 10 frames, in accordance with [52]. For details on the inference runtime, please see *Sup. Mat.*

#### 4.4. Perceptual Study (Comparison to ManipNet)

We evaluate results from CNet and RNet with a perceptual study on Amazon Mechanical Turk (AMT) and compare them with ManipNet and GT motions. For GRAB’s test-set

Metric	ManipNet	GRIP (w/o RNet)	GRIP	Ground truth [51]
Hand-Object Grasp $\uparrow$	$3.68 \pm 1.05$	$4.09 \pm 0.89$	$4.11 \pm 0.85$	$4.12 \pm 0.90$
Hand Motion Smoothness $\uparrow$	$3.8 \pm 0.93$	$3.88 \pm 1.06$	$3.91 \pm 1.04$	$3.98 \pm 1.03$
Contact Consistency $\uparrow$	$3.54 \pm 0.99$	$4.02 \pm 1.01$	$4.09 \pm 0.95$	$4.13 \pm 0.95$
In-Hand Manipulation $\uparrow$	$3.57 \pm 0.99$	$3.96 \pm 1.01$	$3.97 \pm 0.99$	$4.01 \pm 1.00$
Average $\uparrow$	$3.65 \pm 1.00$	$3.99 \pm 1.00$	$4.02 \pm 0.96$	$4.06 \pm 0.97$

Table 1. Perceptual evaluation of GRIP results, without and with RNet, compared with the ManipNet [58] results and ground truth [51]. The participants rate the realism of the generated grasps from 1 (unrealistic) to 5 (very realistic). The table reports the mean  $\pm$  std, computed for all valid study participants. Results show that GRIP generated grasps are more realistic than ManipNet and that RNet improves the grasps of CNet.

motion sequences, we use GRIP to generate the interacting hand poses. We then create videos of the generated motions from CNet, the refined motions from RNet, and the corresponding ground truth. To compare with ManipNet, we extracted their moving meshes from their demo and rendered them in the same format as GRIP results.

The participants rate the realism of the hand motions based on 4 criteria: (1) hand-object grasp, (2) hand motion smoothness, (3) contact consistency, and (4) in-hand manipulations. Each motion is evaluated by at least 10 different participants. The ratings are on a 5-level Likert scale, where 1 means unrealistic and 5 means very realistic. We use a catch trial similar to [51, 52] to identify invalid ratings and remove them; Tab. 1 shows the evaluation results.

Method	MPVPE (mm) ↓		MPJPE (mm) ↓		CC (mm) ↓	
	R-Hand	L-Hand	R-Hand	L-Hand	R-Hand	L-Hand
<b>Hand Sensors Ablation</b>						
GRIP (w/o Ambient)	9.56	6.72	7.08	4.99	15.03	9.48
GRIP (w/o Proximity)	9.62	6.82	7.11	5.09	15.64	9.10
<b>Latent Temporal Consistency (LTC) Evaluation</b>						
GRIP (w/o Consist.)	8.17	6.18	5.99	<b>4.53</b>	13.01	7.66
GRIP (output Consist.)	9.31	7.11	6.81	5.31	13.21	8.18
GRIP (Voxel-grid)	8.36	6.54	6.60	4.75	11.35	6.87
GRIP (w/o RNet)	8.19	6.58	6.10	4.95	11.44	7.03
GRIP (fullmodel)	<b>7.88</b>	<b>6.17</b>	<b>5.85</b>	4.62	<b>10.56</b>	<b>6.25</b>

Table 2. **(Top)** We show the effect of our “Hand Sensors” by comparing variants of GRIP without our sensors’ features; GRIP results in lower errors. **(Bottom)** The effect of the LTC algorithm is explored by comparing GRIP against a network without LTC (*w/o Consist.*) and one with consistency on the output poses (*output Consist.*). The GRIP-generated motions have lower errors.

Metric	Model	GRAB-T (0.01)	GRAB-T (0.02)	GRAB-R (0.3)	GRAB-R (0.5)
MPVPE (mm)	TOCH	16.0 → <b>11.8</b>	31.9 → <b>13.9</b>	<b>6.30</b> → 11.5	<b>10.3</b> → 11.0
	GRIP	17.4 → <b>10.3</b>	34.2 → <b>13.1</b>	6.21 → <b>4.62</b>	10.5 → <b>6.72</b>
MPJPE (mm)	TOCH	16.0 → <b>9.93</b>	31.9 → <b>12.3</b>	<b>4.58</b> → 9.58	<b>7.53</b> → 9.12
	GRIP	16.9 → <b>9.70</b>	33.8 → <b>12.8</b>	4.26 → <b>3.21</b>	7.64 → <b>4.18</b>

Table 3. Comparison of GRIP (ANet & RNet) performance with TOCH [61] on the perturbed test-sets of GRAB. Following TOCH, we perturb the hand pose (-R) and translation (-T) by adding Gaussian noise. The numbers in parentheses (top) show the noise magnitude. Metrics before and after using each method are reported.

The study shows that the GRIP-generated hand motions are very realistic and close to the ground-truth ones. In addition, the scores are slightly higher when motions are refined by RNet, especially for Contact Consistency (CC), which shows the effectiveness of our LTC algorithm. Furthermore, we see a lower rating for ManipNet results compared to our results. Additionally, in Tab. 2 we show the computed penetration errors for ManipNet, which is 13% higher than ours. While the test data is different (simpler for ManipNet), these results confirm several limitations of ManipNet such as single-hand inference, poor generalization to new objects, and no full-body setting. GRIP addresses these issues, making it easy to apply in real-world scenarios. For representative grasps and failures please see *Sup. Mat.*

#### 4.5. Robustness to Noise

To evaluate the performance of ANet and RNet, we compare them to TOCH [61] on refining perturbed test-sets from GRAB. To do this, similar to [61], we perturb the motions by adding Gaussian noise, with different magnitudes, to the pose (GRAB-R) and translation (GRAB-T) of both hands. To keep the original motion dynamics, generated from CNet, RNet is trained to only refine hand-pose (i.e., rotation perturbations), therefore we refine perturbed translation using ANet and perturbed rotations using RNet. We provide the full-comparison results in Tab. 3. Results show that the combination of ANet and RNet performs better in refining noisy hand interactions.

Grasp Model	Penetr. (cm <sup>3</sup> ) ↓	Cont. Ratio ↑	MPJPE (mm) ↓	GRIP		MPVPE (mm) ↓
				# Future Frames	R-Hand	
1. GrabNet [51]	2.65	<b>1.00</b>	<b>4.00</b>	1. 0	9.21	8.18
2. GrabNet-SMPL-X	7.33	0.87	6.72	2. 3	8.94	7.78
3. ManipNet [58]	2.68	0.98	-	3. 5	8.34	7.29
4. GRIP (w/o-RNet)	3.18	0.96	6.34	5. 10	<b>7.88</b>	<b>6.17</b>
5. GRIP (w/ -RNet)	<b>2.38</b>	<b>1.00</b>	5.85			
GRAB (GT)	1.95	1.00	-			

Table 4. **(Left)** Penetration and contact-ratio metrics for two GrabNet baselines and GRIP models. **(Right)** Trade-off between the latency (see number # of future frames) and accuracy of GRIP.

#### 4.6. Grasp Evaluation

To evaluate grasps, Tab. 4-left reports the penetration volume (cm<sup>3</sup>), contact ratio [59], and MPJPE errors for: (1) *GrabNet* [51], which generates MANO grasps, (2) a trained *GrabNet-SMPL-X* variant, which generates full-body SMPL-X grasps, (3) *ManipNet* [58], (4) *GRIP (w/o RNet)*, and (5) *GRIP (w/ RNet)*. Results show that our full model (row 5) outperforms baselines by generating grasps with less penetrations and better contact. However, GrabNet produces lower MPJPE. This is because GRIP generates the motion of *both* hands for the full *dynamic* interaction, i.e., before, during and after object manipulation. Instead, GrabNet variants only generate *static* grasps of *one* hand. We argue that hand poses for *non-grasping frames* can deviate from ground-truth ones, as the object does not constrain poses, thus, GRIP has a higher MPJPE.

### 5. Conclusion

We present GRIP, a learned method that generates realistic motions for both hands given an animated body and object in interaction. Our novelties include: (1) a coarse-to-fine method for dynamic grasp generation, (2) a set of distance sensors, and (3) a latent-space temporal consistency. As a result, GRIP is able to refine noisy arm motions and predict hand poses from scratch, handle novel object shapes, adapt to bi-manual interactions, and generate temporally-consistent hand motion. GRIP can help capture new data of human-object interaction *without* the difficulty of tracking hands, add hands to previous datasets [30, 38], and synthesize hands for avatars in video games, movies, and AR/VR. **Limitations & Future Work:** GRIP uses 10 future frames to guide grasp prediction, causing a 10-frame delay; motion anticipation might reduce delays [53]. Future work can extend GRIP to human-scene interaction [21], remove noise for the full body and object, instead of just the arms, and apply RNet on bodies recovered from images [12, 25].

**Acknowledgements:** This work was partially supported by Adobe Research (OT’s internship), the International Max Planck Research School for Intelligent Systems (IMPRS-IS), and the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039B. We thank T. Alexiadis and A. Cseke for perceptual studies.

**Disclosure:** [https://files.is.tue.mpg.de/black/CoL\\_3DV\\_2024.txt](https://files.is.tue.mpg.de/black/CoL_3DV_2024.txt)

## References

- [1] Rami Ali Al-Asqhar, Taku Komura, and Myung Geol Choi. Relationship descriptors for interactive motion adaptation. In *Symposium on Computer Animation (SCA)*, pages 45–53, 2013. 3
- [2] Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Józefowicz, Bob McGrew, Jakub W. Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation. *International Journal of Robotics Research (IJRR)*, 39(1), 2020. 3
- [3] Kevin Bergamin, Simon Clavet, Daniel Holden, and James Richard Forbes. DReCon: Data-driven responsive control of physics-based characters. *Transactions on Graphics (TOG)*, 38(6), 2019. 3
- [4] Bharat Lal Bhatnagar, Xianghui Xie, Ilya A Petrov, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Behave: Dataset and method for tracking human object interactions. In *Computer Vision and Pattern Recognition (CVPR)*, pages 15935–15946, 2022. 3
- [5] Jeannette Bohg, Antonio Morales, Tamim Asfour, and Danica Kragic. Data-driven grasp synthesis—a survey. *IEEE Transactions on Robotics*, 30:289–309, 2014. 3
- [6] Samarth Brahmabhatt, Cusuh Ham, Charles C. Kemp, and James Hays. ContactDB: Analyzing and predicting grasp contact via thermal imaging. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 3
- [7] Samarth Brahmabhatt, Chengcheng Tang, Christopher D. Twigg, Charles C. Kemp, and James Hays. ContactPose: A dataset of grasps with object contact and hand pose. In *European Conference on Computer Vision (ECCV)*, pages 361–378, 2020. 3
- [8] Yu-Wei Chao, Jimei Yang, Weifeng Chen, and Jia Deng. Learning to sit: Synthesizing human-chair interactions via hierarchical control. In *Conference on Artificial Intelligence (AAAI)*, pages 5887–5895, 2021. 3
- [9] Tao Chen, Jie Xu, and Pulkit Agrawal. A system for general in-hand object re-orientation. *Conference on Robot Learning (CoRL)*, 2021. 3
- [10] Sammy Christen, Muhammed Kocabas, Emre Aksan, Jemin Hwangbo, Jie Song, and Otmar Hilliges. D-Grasp: Physically plausible dynamic grasp synthesis for hand-object interactions. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 3
- [11] Enric Corona, Albert Pumarola, Guillem Alenyà, Francesc Moreno-Noguer, and Gregory Rogez. GanHand: Predicting human grasp affordances in multi-object scenes. In *Computer Vision and Pattern Recognition (CVPR)*, pages 5030–5040, 2020. 3
- [12] Sai Kumar Dwivedi, Cordelia Schmid, Hongwei Yi, Michael J. Black, and Dimitrios Tzionas. POCO: 3D pose and shape estimation using confidence. In *International Conference on 3D Vision (3DV)*, 2024. 8
- [13] Sahar El-Khoury, Anis Sahbani, and Philippe Bidaud. 3D objects grasps synthesis: A survey. In *IFTOMM World Congress on Mechanism and Machine Science*, 2011. 3
- [14] 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 *Computer Vision and Pattern Recognition (CVPR)*, pages 12943–12954, 2023. 3
- [15] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with RGB-D videos and 3D hand pose annotations. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 3
- [16] Guillermo Garcia-Hernando, Edward Johns, and Tae-Kyun Kim. Physics-based dexterous manipulations with estimated hand poses and residual reinforcement learning. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 9561–9568, 2020. 3
- [17] Michael Gleicher. Retargetting motion to new characters. In *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 33–42, 1998. 3
- [18] Patrick Grady, Chengcheng Tang, Christopher D. Twigg, Minh Vo, Samarth Brahmabhatt, and Charles C. Kemp. ContactOpt: Optimizing contact to improve grasps. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1471–1481, 2021. 2, 3
- [19] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3193–3203, 2020. 3
- [20] Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael J. Black. Stochastic scene-aware motion prediction. In *International Conference on Computer Vision (ICCV)*, pages 11374–11384, 2021. 3
- [21] Mohamed Hassan, Partha Ghosh, Joachim Tesch, Dimitrios Tzionas, and Michael J. Black. Populating 3D scenes by learning human-scene interaction. In *Computer Vision and Pattern Recognition (CVPR)*, pages 14708–14718, 2021. 8
- [22] Yana Hasson, Gül Varol, Dimitris Tzionas, Igor Kalevatykh, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Computer Vision and Pattern Recognition (CVPR)*, pages 11807–11816, 2019. 3
- [23] Yana Hasson, Bugra Tekin, Federica Bogo, Ivan Laptev, Marc Pollefeys, and Cordelia Schmid. Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In *Computer Vision and Pattern Recognition (CVPR)*, pages 568–577, 2020. 3
- [24] Edmond S. L. Ho, Taku Komura, and Chiew-Lan Tai. Spatial relationship preserving character motion adaptation. *Transactions on Graphics (TOG)*, 29(4):33:1–33:8, 2010. 3
- [25] Yinghao Huang, Omid Taheri, Michael J. Black, and Dimitrios Tzionas. InterCap: Joint markerless 3D tracking of humans and objects in interaction. In *German Conference on Pattern Recognition (GCPR)*, pages 281–299, 2022. 1, 2, 3, 4, 6, 8
- [26] Hanwen Jiang, Shaowei Liu, Jiashun Wang, and Xiaolong Wang. Hand-object contact consistency reasoning for human grasps generation. In *International Conference on Computer Vision (ICCV)*, pages 11087–11096, 2021. 3
- [27] Mubbasir Kapadia, Xu Xianghao, Maurizio Nitti, Marcelo Kallmann, Stelian Coros, Robert W. Sumner, and Markus

- Gross. Precision: Precomputing environment semantics for contact-rich character animation. In *Symposium on Interactive 3D Graphics (SI3D)*, 2016. 3
- [28] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J. Black, Krikamol Muandet, and Siyu Tang. Grasping Field: Learning implicit representations for human grasps. In *International Conference on 3D Vision (3DV)*, pages 333–344, 2020. 2, 3
- [29] Korrawe Karunratanakul, Adrian Spurr, Zicong Fan, Otmar Hilliges, and Siyu Tang. A skeleton-driven neural occupancy representation for articulated hands. In *International Conference on 3D Vision (3DV)*, pages 11–21, 2021. 3
- [30] Philipp Kratzer, Simon Bihlmaier, Niteesh Balachandra Midlagajni, Rohit Prakash, Marc Toussaint, and Jim Mainprice. MoGaze: A dataset of full-body motions that includes workspace geometry and eye-gaze. *Robotics and Automation Letters (RA-L)*, 6(2):367–373, 2021. 3, 6, 8
- [31] Paul G. Kry and Dinesh K. Pai. Interaction capture and synthesis. *Transactions on Graphics (TOG)*, 25(3):872–880, 2006. 3
- [32] Jehee Lee, Jinxiang Chai, Paul S. A. Reitsma, Jessica K. Hodgins, and Nancy S. Pollard. Interactive control of avatars animated with human motion data. *Transactions on Graphics (TOG)*, 21(3):491–500, 2002. 3
- [33] Kang Hoon Lee, Myung Geol Choi, and Jehee Lee. Motion patches: Building blocks for virtual environments annotated with motion data. *Transactions on Graphics (TOG)*, 25(3):898–906, 2006. 3
- [34] Beatriz León, Stefan Ulbrich, Rosen Diankov, Gustavo Puche, Markus Przybylski, Antonio Morales, Tamim Asfour, Sami Moisió, Jeannette Bohg, and James Kuffner. Open-grasp: A toolkit for robot grasping simulation. In *Simulation, Modeling, and Programming for Autonomous Robots SIMPAR*, pages 109–120, 2010. 3
- [35] Ying Li, Jiaxin L. Fu, and Nancy S. Pollard. Data-driven grasp synthesis using shape matching and task-based pruning. *Transactions on Visualization and Computer Graphics (TVCG)*, 13(4):732–747, 2007. 3
- [36] Karen C. Liu. Dexterous manipulation from a grasping pose. *Transactions on Graphics (TOG)*, 28(3):59, 2009. 3
- [37] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A Skinned Multi-Person Linear Model. *Transactions on Graphics (TOG)*, 34(6):248:1–248:16, 2015. 3
- [38] Christian Mandery, Ömer Terlemez, Martin Do, Nikolaus Vahrenkamp, and Tamim Asfour. The KIT whole-body human motion database. In *International Conference on Advanced Robotics (ICAR)*, pages 329–336, 2015. 3, 8
- [39] Igor Mordatch, Zoran Popovic, and Emanuel Todorov. Contact-invariant optimization for hand manipulation. In *Symposium on Computer Animation (SCA)*, pages 137–144, 2012. 3
- [40] Soohwan Park, Hoseok Ryu, Seyoung Lee, Sunmin Lee, and Jehee Lee. Learning predict-and-simulate policies from unorganized human motion data. *Transactions on Graphics (TOG)*, 38(6), 2019. 3
- [41] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019. 3, 4, 5
- [42] Xue Bin Peng, Glen Berseth, and Michiel Van de Panne. Terrain-adaptive locomotion skills using deep reinforcement learning. *Transactions on Graphics (TOG)*, 35(4):81:1–81:12, 2016. 3
- [43] Xue Bin Peng, Glen Berseth, Kangkang Yin, and Michiel Van De Panne. DeepLoco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *Transactions on Graphics (TOG)*, 36(4), 2017. 3
- [44] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. DeepMimic: Example-guided deep reinforcement learning of physics-based character skills. *Transactions on Graphics (TOG)*, 37(4):143:1–143:14, 2018. 3
- [45] Sören Pirk, Olga Diamanti, Boris Thibert, Danfei Xu, and Leonidas J. Guibas. Shape-aware spatio-temporal descriptors for interaction classification. In *International Conference on Image Processing (ICIP)*, pages 4527–4531, 2017. 3
- [46] Sören Pirk, Vojtech Krs, Kaimo Hu, Suren Deepak Rajasekaran, Hao Kang, Yusuke Yoshiyasu, Bedrich Benes, and Leonidas J. Guibas. Understanding and exploiting object interaction landscapes. *Transactions on Graphics (TOG)*, 36(3):31:1–31:14, 2017. 3
- [47] Nancy S. Pollard and Victor Brian Zordan. Physically based grasping control from example. In *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 311–318, 2005. 3
- [48] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Robotics: Science and Systems (RSS)*, 2018. 3
- [49] Qijin She, Ruizhen Hu, Juzhan Xu, Min Liu, Kai Xu, and Hui Huang. Learning high-dof reaching-and-grasping via dynamic representation of gripper-object interaction. *Transactions on Graphics (TOG)*, 41(4), 2022. 3
- [50] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *Transactions on Graphics (TOG)*, 38(6):209:1–209:14, 2019. 2, 4
- [51] Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *European Conference on Computer Vision (ECCV)*, pages 581–600, 2020. 1, 2, 3, 7, 8
- [52] Omid Taheri, Vasileios Choutas, Michael J. Black, and Dimitrios Tzionas. GOAL: Generating 4D whole-body motion for hand-object grasping. In *Computer Vision and Pattern Recognition (CVPR)*, pages 13253–13263, 2022. 2, 3, 4, 5, 7
- [53] Julian Tanke, Chintan Zaveri, and Juergen Gall. Intention-based long-term human motion anticipation. In *International Conference on 3D Vision (3DV)*, pages 596–605, 2021. 8
- [54] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics sim-

- ulation. *International Journal of Computer Vision (IJCV)*, 118(2):172–193, 2016. [3](#)
- [55] He Wang, Sören Pirk, Ersin Yumer, Vladimir G. Kim, Ozan Sener, Srinath Sridhar, and Leonidas J. Guibas. Learning a generative model for multi-step human-object interactions from videos. *Computer Graphics Forum (CGF)*, 38(2):367–378, 2019. [3](#)
- [56] Jiashun Wang, Huazhe Xu, Jingwei Xu, Sifei Liu, and Xiaolong Wang. Synthesizing long-term 3D human motion and interaction in 3D scenes. In *Computer Vision and Pattern Recognition (CVPR)*, pages 9401–9411, 2021. [3](#)
- [57] Yuting Ye and C. Karen Liu. Synthesis of detailed hand manipulations using contact sampling. *Transactions on Graphics (TOG)*, 31(4):41:1–41:10, 2012. [3](#)
- [58] He Zhang, Yuting Ye, Takaaki Shiratori, and Taku Komura. ManipNet: Neural manipulation synthesis with a hand-object spatial representation. *Transactions on Graphics (TOG)*, 40(4):121:1–121:14, 2021. [2](#), [3](#), [4](#), [7](#), [8](#)
- [59] Yan Zhang, Mohamed Hassan, Heiko Neumann, Michael J. Black, and Siyu Tang. Generating 3D people in scenes without people. In *Computer Vision and Pattern Recognition (CVPR)*, pages 6193–6203, 2020. [8](#)
- [60] Wenping Zhao, Jianjie Zhang, Jianyuan Min, and Jinxiang Chai. Robust realtime physics-based motion control for human grasping. *Transactions on Graphics (TOG)*, 32(6):207:1–207:12, 2013. [3](#)
- [61] Keyang Zhou, Bharat Lal Bhatnagar, Jan Eric Lenssen, and Gerard Pons-Moll. TOCH: Spatio-temporal object correspondence to hand for motion refinement. In *European Conference on Computer Vision (ECCV)*, pages 1–19, 2022. [2](#), [3](#), [8](#)
- [62] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Computer Vision and Pattern Recognition (CVPR)*, pages 5745–5753, 2019. [4](#)
- [63] Tianqiang Zhu, Rina Wu, Xiangbo Lin, and Yi Sun. Toward human-like grasp: Dexterous grasping via semantic representation of object-hand. In *International Conference on Computer Vision (ICCV)*, pages 15721–15731, 2021. [3](#)