3D Whole-Body Grasp Synthesis with Directional Controllability

Georgios Paschalidis[1] Romana Wilschut[1] Dimitrije Antić[1] Omid Taheri[2] Dimitrios Tzionas[1]
[1]University of Amsterdam, the Netherlands [2]Max Planck Institute for Intelligent Systems, Tübingen, Germany
{g.paschalidis, d.antic, d.tzionas}@uva.nl romana.wilschut@gmail.com otaheri@tue.mpg.de

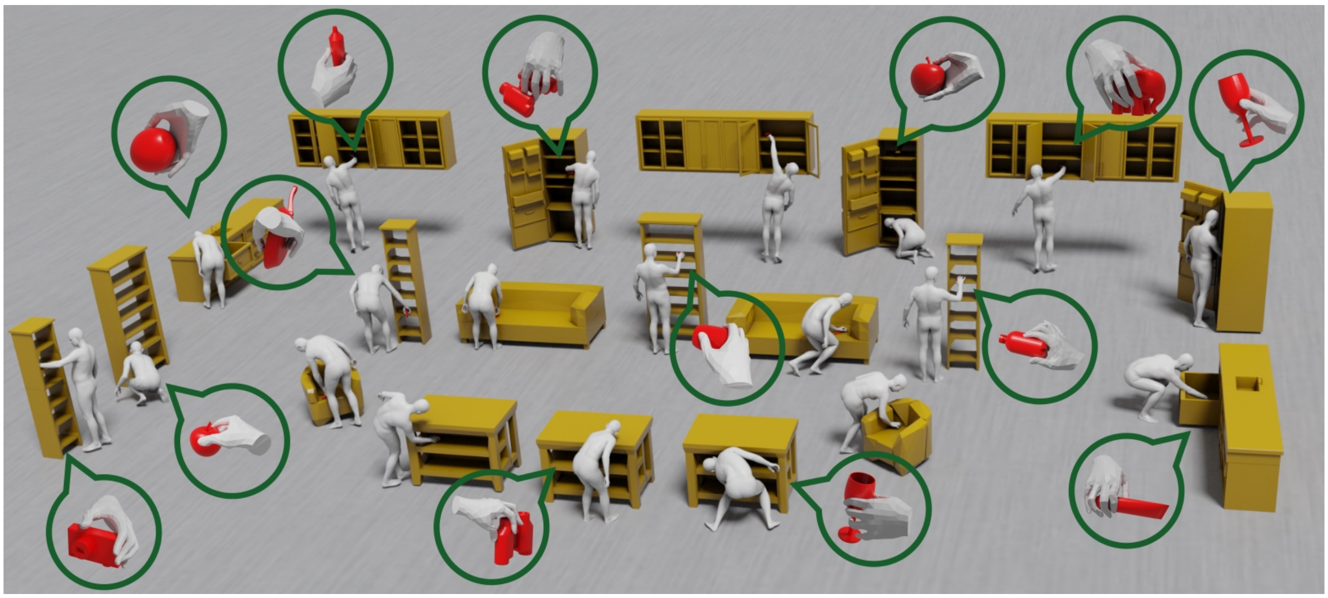

Figure 1. We develop CWGrasp, a novel framework for synthesizing 3D whole-body grasps for an object placed on a receptacle. Our framework builds on a novel combination of geometric-based reasoning and controllable data-driven synthesis methods. By adding a novel controllability in the synthesis process, we achieve realistic results at a fraction of the computational cost w.r.t. the state of the art [54].

Abstract

Synthesizing 3D whole bodies that realistically grasp objects is useful for animation, mixed reality, and robotics. This is challenging, because the hands and body need to look natural w.r.t. each other, the grasped object, as well as the local scene (i.e., a receptacle supporting the object). Moreover, training data for this task is really scarce, while capturing new data is expensive. Recent work goes beyond finite datasets via a divide-and-conquer approach; it first generates a "guiding" right-hand grasp, and then searches for bodies that match this. However, the guiding-hand synthesis lacks controllability and receptacle awareness, so it likely has an implausible direction (i.e., a body can't match this without penetrating the receptacle) and needs corrections through major post-processing. Moreover, the body search needs exhaustive sampling and is expensive. These are strong limitations. We tackle these with a novel method called CWGrasp. Our key idea is that performing geometry-based reasoning "early on," instead of "too late," provides rich "control" signals for inference. To this end, CWGrasp first samples a plausible reaching-direction vector (used later for both the arm and hand) from a probabilistic model built via ray-casting from the object and collision checking. Then, it generates a reaching body with a desired arm direction, as well as a "guiding" grasping hand with a desired palm direction that complies with the arm's one. Eventually, CWGrasp refines the body to match the "guiding" hand, while plausibly contacting the scene. Notably, generating already-compatible "parts" greatly simplifies the "whole". Moreover, CWGrasp uniquely tackles both right- and left-hand grasps. We evaluate on the GRAB and ReplicaGrasp datasets. CWGrasp outperforms baselines, at lower runtime and budget, while all components help performance. Code and models are available at https://gpaschalidis.github.io/cwgrasp.

1. Introduction

Synthesizing virtual 3D humans that grasp objects realistically is important for applications such as virtual assistants, animation, robotics, games, or synthetic image datasets. Importantly, this involves the whole body, so that the body approaches an object, arms reach it, and hands grasp it. But this is challenging; the body and hands should look natural and fully coordinated, the body should approach an object without penetrating the scene, the hands should dexterously contact the object. Due to these challenges, most of the existing work tackles only parts of the problem, namely disembodied hands, or bodies with non-dexterous hands.

To make matters worse, 3D training data for whole-body grasps is very scarce. The recent FLEX [54] method tackles data scarcity in a divide-and-conquer way. First, it generates a hand-only grasp through GrabNet [50]. Then, this grasping hand guides a search for a plausible body. That is, many bodies are sampled in random poses and locations, and are optimized to match the guiding hand. However, there exists a key problem; the guiding hand has a random direction that likely disagrees with the direction bodies can approach from without penetrating receptacles. So, the guiding hand needs major corrections via post-processing. This produces promising results but needs exhaustive sampling (500 bodies), and is expensive (separate refinement per sample).

We identify two main reasons for the above problems: (1) Performing body- and receptacle-aware reasoning "too late", and (2) GrabNet's total lack of controllability[1]. These are key limitations. We tackle these by developing **CWGrasp** ("Controllable Whole-body Grasp synthesis"), a new method composed of the following novel modules.

ReachingField model: First, we detect the directions from which a body's arm and hand can reach an object without penetrating the receptacle supporting the object. Think of a mug lying on a shelf and emitting "light"; some rays travel unblocked in free space, while other ones get blocked by shelf panels. Our key insight is that the "well-lit" space near the object reveals its reachability. So, we cast rays from the object, detect collisions with nearby receptacles, and consider only the non-colliding rays for building a new probabilistic 3D vector field, called ReachingField.

Sampling the ReachingField provides a single 3D direction vector that can be used as a "control signal" for the synthesis of both a reaching body and grasping hand. But existing synthesizers for this, such as GNet [51] for the body and GrabNet [50] for the hand, lack such controllability[1,2]. We resolve this with two novel modules, as follows.

[1] GrabNet [50] uses wrist translation and rotation only for training. For inference the only input is object shape, so grasps have a random direction.

[2] GNet [51] takes as input only object shape and height.

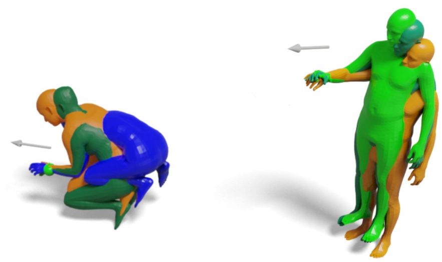

Figure 2. **Controllable reaching-body synthesis (CReach).** We show examples where multiple bodies (shown with several colors) are generated to reach a target wrist location (shown as a green sphere), while having a *desired 3D arm direction* (gray arrow).

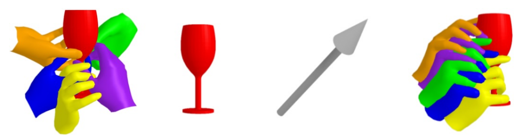

Figure 3. **Controllable hand-grasp synthesis.** The goal is to grasp the red wineglass. **Left – GrabNet [50]:** Due to GrabNet's lack of controllability[1], sampling its latent space produces plausible grasps (shown with several colors) but with *random direction*. **Right – Our CGrasp:** We add controllability, so drawing samples produces plausible and varied grasps (shown with several colors), that have a *desired 3D palm direction* (shown with a gray arrow).

CReach model: We train a conditional variational autoencoder (cVAE) for producing a reaching SMPL-X [44] body. This goes beyond GNet in three ways: (1) It is conditioned not only on target object/wrist location, but, uniquely, also on a desired 3D arm direction; see Fig. 2. (2) It is trained not only on GRAB [50] data, which has a limited range of target wrist (and object) locations, but also on CIRCLE [3] data that is richer for reaching body poses. (3) It can generate both right- and left-arm reaching. We call the resulting model CReach for "Controllable Reach."

CGrasp model: We train a cVAE to generate a grasping MANO [46] hand. This goes beyond GrabNet in two ways: (1) It is conditioned not only on object shape[1], but also on a desired 3D palm direction; see examples in Fig. 3-right. (2) It can generate both right- and left-hand grasps. We call the resulting model CGrasp for "Controllable Grasp."

CWGrasp framework: We condition *both* CReach and CGrasp on the *same* direction, produced by ReachingField. Crucially, this produces a reaching SMPL-X body (CReach) and a guiding MANO grasping hand (CGrasp) that are *already "compatible"* with each other, so they only need a small refinement to be "put together." To this end, we conduct optimization [26, 44, 54] that searches for the SMPL-X pose that lets SMPL-X's hand match the guiding MANO hand, while the body contacts the floor without penetrating the receptacle. Thanks to our controllable inference, we can sample only 1 body and hand from CReach and CGrasp, respectively, in strong contrast to FLEX's 500 different samples. This makes our framework roughly $16\times$ faster.

We evaluate on the GRAB [50] and ReplicaGrasp [54] datasets. Both CGrasp and CReach accurately preserve a specified palm and arm direction, respectively. Importantly, adding controllability does not harm; CGrasp performs on par with three baselines [36, 50, 58] while being able to control palm direction. Last, our CWGrasp method outperforms FLEX [54] in almost all metrics, while its generated whole-body grasps are perceived as more realistic.

In summary, here we make four main contributions:

1. The *ReachingField* model that generates 3D directions for reaching a 3D object, helping as a control signal.
2. The *CReach* model that generates a SMPL-X body reaching objects with a desired (right/left) arm direction.
3. The *CGrasp* model that generates a (right/left) MANO hand grasping an object with a desired palm direction.
4. The novel *CWGrasp* method that combines the above for generating dexterous SMPL-X grasps for an object lying on a receptacle. This is 16x faster than a SotA baseline, and uniquely tackles both right- and left-hand grasps.

2. Related Work

2.1. Hand-only Grasps

Early research focused on modeling [14, 41] and classifying [13, 19] grasps. Then, research focused on generating grasps for robot [5, 32] and human hands [6, 7, 50].

Hand models: Some work models hand shape explicitly with 3D meshes [4, 42] with statistical models [35, 46] being popular. Other work uses implicit shape, such as 3D distance fields [12, 30] or a sum of 3D Gaussians [47]. Here we use MANO [46] due its wide user base, and because it lets us compute accurate contacts and penetrations.

Data: Many datasets have been captured with single- [8, 9, 20, 24, 38, 69] or two-hand [25, 33] images. Recent work captures whole-body meshes [44] grasping rigid objects [50], or articulated objects while also containing RGB images [16]. HOIDiffusion [63] uses a diffusion model for generating synthetic hand-object images conditioned on 3D hand-object grasps produced by GrabNet [50]. DexGraspNet [58] builds a large dataset by applying an optimization framework on 3D objects, leveraging a differentiable force closure estimator and energy functions. Here we extend the GrabNet [50] model and use its GRAB [50] dataset to train our model to facilitate fair comparisons.

Contact: ContactGrasp [7] uses real contact maps from the ContactDB [6] dataset to infer a grasping hand pose, given a posed object mesh. ContactOpt [23] infers likely hand-object contacts and optimizes hand pose to match these. GraspTTA [29] infers an initial grasp for an object point cloud, and optimizes it to match a target contact map. Grasp'D [56] takes a hand, an object as a point cloud and as a SDF, and generates grasps via optimization on contact forces. ContactGen [36] learns an object-conditioned joint distribution of a contact-, part- and direction-map, exploiting the direction of contact at a low level for synthesis. GrabNet [50] infers an initial grasp for a BPS-encoded [45] object and refines it with a neural net that considers a per-vertex contact likelihood. GrabNet lacks controllability, so it produces grasps with random directions. Here we extend GrabNet by adding the missing controllability; only our and concurrent work [62] condition on the palm's direction.

Grasps from images: ObMan [28] infers hand and object meshes from a color image, while H+O [53] infers keypoints. GanHand [11] infers object pose and grasp type with a rough hand pose [19], refining it via contact constraints. TOCH [67] does a refinement using a 3D SDF. More recent work tackles grasps with unknown objects from color video [18, 49]. For a more detailed overview please see [17].

Motion generation: D-Grasp [10] learns hand-object interaction via RL; the task is to grasp and move a given object to a goal pose. ManipNet [61] generates hand-object interaction (HOI) motions for single or both hands, using spatial features. GeneOH Diffusion [37] denoises HOI motion via diffusion, and a hand-keypoint trajectory representation. GRIP [52] and GEARS [68] synthesize interacting finger motion from given hand and object trajectories. Concurrently to us, GraspXL [62] generates grasping motions via RL (without using pre-captured HOI data) while conditioning on the palm direction, as we do for static synthesis.

2.2. Whole-body Grasps

The shape representation used for body models ranges from cylinders [40] and super-quadrics [21] to mesh-based statistical 3D models [1, 2, 39, 43, 44, 60]. We use the SMPL-X [44] statistical model that is widely used for interactions.

Interacting with scenes: Wang et al. [57] first infer intermediate key poses and then generate in-between motions. SAMP [27] and NSM [48] infer several goal locations and orientations on target objects, (stochastically and deterministically, respectively), and then infer in-between motion. Given a body pose and chair mesh, COUCH [65] infers diverse contacts on the chair, and body poses that match these.

Static grasps: FLEX [54] generates SMPL-X grasps, by optimizing the body to match a guiding hand-grasp inferred via GrabNet [50]. Our CWGrasp method is inspired by this, but is more efficient thanks to its controllable inference.

Dynamic grasps: CIRCLE [3] and WANDR [15] infer (short- and long-term, respectively) motion for reaching a target wrist location. GOAL [51] infers a static target body grasp via interaction-aware features, and infers motion to the goal. SAGA [59] generates such motions stochastically. IMoS [22] infers guiding arm-only motions that "drive" the whole body. Given object trajectories, OMOMO [34] uses a conditional denoising diffusion model to generate wrist joint positions for each object state, and then conditions on these to generate a full body with non-articulated hands.

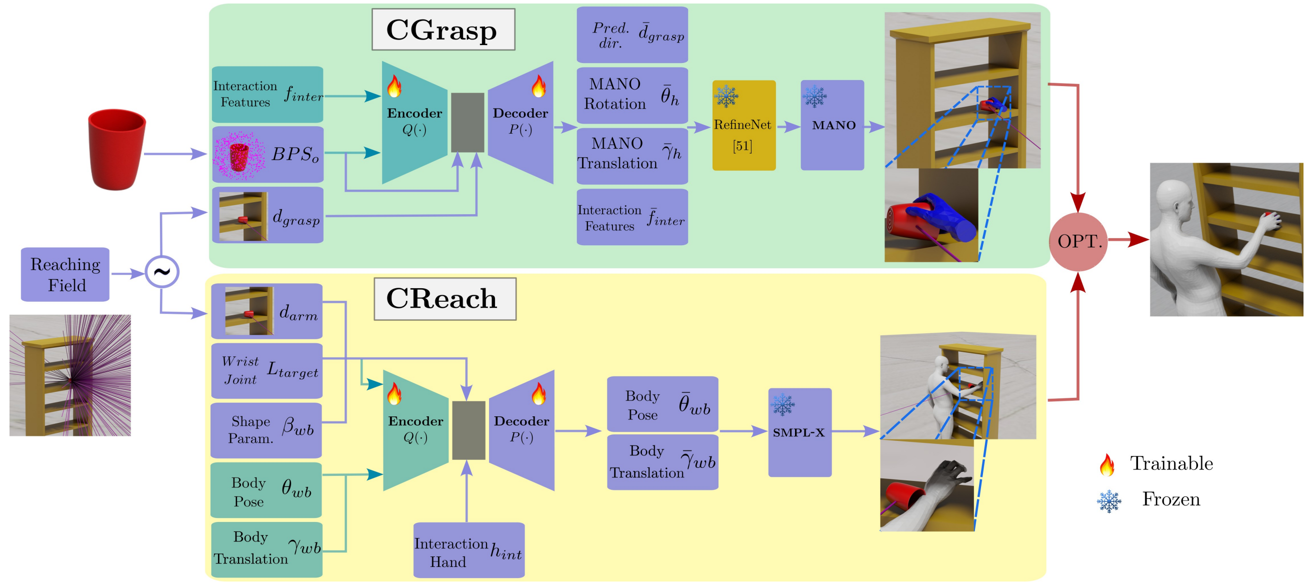

Figure 4. **CWGrasp framework.** We first sample a single reaching direction from ReachingField. Next, we condition both CGrasp and CReach on the same direction and obtain a guiding hand grasp (shown in blue) and a reaching body (shown in gray), respectively, that satisfy the sampled direction, so they are "compatible" with each other. Finally, an optimization stage refines the body to match the guiding hand while resolving penetrations with the object and/or receptacle. Note that our framework can generate both left- and right-hand grasps. Parts in purple are used for both training and inference, in green only for training, in brown only for inference, and in red for optimization.

3. Method

We build CWGrasp, a novel framework (Fig. 4) that generates a whole-body grasp, given an object on a receptacle. To this end, we develop ReachingField (Sec. 3.2), a novel model that generates a likely reaching direction. We condition on the same direction two novel models for producing a reaching body (CReach, Sec. 3.3) and hand grasp (CGrasp, Sec. 3.4). We combine all these in CWGrasp (Sec. 3.5).

3.1. Preliminaries

Hand model (used in CGrasp): We use MANO [46], a differentiable function $M_h(\beta_h, \theta_h, \gamma_h)$ parameterized by translation, $\gamma_h \in \mathbb{R}^3$, shape, $\beta_h \in \mathbb{R}^{10}$, and pose, θ_h. The output is a 3D mesh, M_h, rigged with a skeleton of 16 joints; 1 for the wrist and 15 for fingers. The pose $\theta_h \in \mathbb{R}^{16\times3}$ is encoded as axis-angle rotations; the global rotation (first 3 parameters) is $\theta_h^{\text{wrist}} \in \mathbb{R}^3$. The shape parameters β_h live in a low-dimensional linear space.

Whole-body model (used in CReach, CWGrasp): We use the SMPL-X [44] model, a differentiable function $M_{wb}(\beta_{wb}, \theta_{wb}, \gamma_{wb})$ parameterized by shape, $\beta_{wb} \in \mathbb{R}^{10}$, pose, θ_{wb}, and translation, $\gamma_{wb} \in \mathbb{R}^3$; here we ignore facial parameters. The output is a 3D mesh, M_{wb}, rigged with a skeleton of 22 body joints and 15 joints per hand. The pose $\theta = (\theta_b, \theta_h)$ consists of $\theta_b \in \mathbb{R}^{22\times3}$ for the body and $\theta_h \in \mathbb{R}^{2\times15\times3}$ for hands as axis-angle rotations. The shape parameters β_{wb} live in a low-dimensional linear space.

CoarseNet – part of GrabNet [50]: GrabNet generates 3D MANO grasps for a given object, and consists of: CoarseNet, for producing an initial grasp, and RefineNet, for refining it. Here we focus only on grasp generation, so we build on **CoarseNet**. This is modeled as a VAE; given an object shape represented with Basis Point Sets [45], BPS_o, a wrist rotation, θ_h^{wrist}, and translation, γ_h, the encoder Q generates a latent code $Z \in \mathbb{R}^{16}$, namely: $Q(Z|\theta_h^{\text{wrist}}, \gamma_h, BPS_o)$. The decoder maps this, concatenated with the object shape, BPS_o, to an estimated MANO translation, $\bar{\gamma}_h \in \mathbb{R}^3$, and joint angles, $\bar{\theta}_h \in \mathbb{R}^{16\times6}$, i.e.: $P(\bar{\theta}_h, \bar{\gamma}_h|Z, BPS_o)$. To train CoarseNet we use both its encoder and decoder, and use 5 losses: $\mathcal{L}_{KL}$, $\mathcal{L}_{edge}$, $\mathcal{L}_{vertex}$, $\mathcal{L}_{d_{o2h}}$, $\mathcal{L}_{d_{h2o}}$; for details see [50]. In test time, we use only the decoder conditioned on object shape, BPS_o; there is no other input. Thus, sampling different latent codes produces grasping hands with a random direction; see Fig. 3.

GNet – part of GOAL [51]: GNet generates a SMPL-X grasping body for a given object shape and location. GNet is modeled with a VAE, like CoarseNet, so it has an encoder, $Q(Z|\theta_{wb}, \gamma_{wb}, L_{\text{target}})$, and decoder, $P(\bar{\theta}_{wb}, \bar{\gamma}_{wb}|Z, L_{\text{target}})$, where Z is the latent code, θ_{wb} is body pose, γ_{wb} is translation. L_{target} is a target condition comprising the object's shape, BPS_o, and its centroid height. For details see [51].

3.2. ReachingField – Reaching Direction

Given an object on a receptacle, we build ReachingField, a novel probabilistic 3D vector field of directions the object can be reached by a body (see Fig. 5). To this end, we cast 3D rays from the object to surrounding space, check for collisions with the receptacle, filter out colliding ones (considering also the arm's volume and standing on the ground), and assign probabilities to remaining rays, as follows.

Ray casting: Let $\mathcal{O}$ be a 3D mesh for the object, and $\mathbf{c} \in \mathbb{R}^3$ be its centroid. We sample uniformly a (unit) sphere centered at $\mathbf{c}$, constructing a spherical point grid $\mathbf{S} = \{s_i\}$. Then, we cast rays r_i going from $\mathbf{c}$ through each point s_i.

Ray filtering: Let $\mathcal{M}$ be a 3D mesh for the receptacle. We evaluate and filter the casted rays r_i with the following. **Filter #1. Arm/hand direction (Fig. 5):** We traverse each ray r_i and evaluate whether it intersects with $\mathcal{M}$. Intersecting rays are pruned, as they represent a direction from which an arm or hand would "directly" penetrate the receptacle. **Filter #2 - Body orientation (Fig. 6):** To (optionally) save computational resources (on the expense of pruning some plausible directions), we project the curated rays onto a horizontal plane parallel to the ground, and detect further intersections with $\mathcal{M}$. Intersecting rays denote directions that hinder a body from "easily" approaching the object. However, in case all rays intersect, e.g. when the object is inside a box or drawer, then this step is disregarded altogether. **Filter #3 - Standing places:** To grasp an object, a body needs to stand at a nearby place on the ground without penetrating any "occluders." To find such places, we traverse the curated rays r_i, and at regular intervals (every 30 cm) we cast vertical rays r_{ij} and check whether these collide with $\mathcal{M}$ or other "occluders" hindering a body from standing. In case of collision we prune the "parent" ray r_i altogether. **Filter #4 - Wiggle room for arm volume:** The above steps "detect" plausible rough body positions and arm directions. However, they ignore that a body has a certain *volume*, so its vertices can still penetrate the receptacle. To resolve this, we "swipe" all projected filtered rays within a small range around the vertical axis, and discard those intersecting $\mathcal{M}$.

ReachingField: The curated rays are plausible reaching directions. But not all directions are equally likely. When changing a light bulb on the ceiling, our hand likely approaches it from below, while when tying shoelaces, it approaches from above. Thus, likelihood depends on how high above the ground an object lies and is defined as:

$$p_i = \frac{\exp\left(-1/(s_i a_i)\right)}{\sum_i \exp\left(-1/(s_i a_i)\right)}, \tag{1}$$

where a_i is the smallest angle of ray r_i w.r.t. the vertical axis z, while $s_i = -1$ when the object height is $\geq$0.7m above ground and r_i is directed downward, or the height is $<$0.7m and r_i is directed upward. Else, $s_i = 1$. See likelihood examples in Fig. 7. For details see Sup. Mat. (Sec. S.1.1).

Inference: ReachingField is probabilistic, so sampling it produces a plausible 3D reaching direction. Note that objects can be reached from multiple directions; drawing different samples accounts for this.

3.3. CReach – Controllable Reaching Bodies

Our goal is *controllable* synthesis of a SMPL-X body "reaching" an object. We do this by extending GNet [51] with a *condition* on *arm direction*; see Fig. 4 bottom.

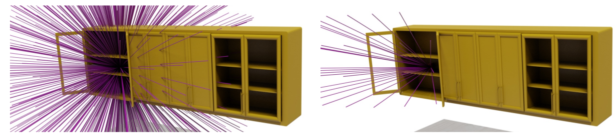

Figure 5. **Arm/hand direction (Sec. 3.2, Filter #1)**. **Left:** We cast rays from the object to surrounding space. **Right:** We prune rays intersecting with a receptacle and keep non-intersecting ones; the latter represent directions an arm/hand can reach the object from.

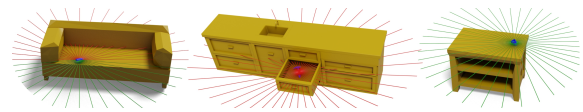

Figure 6. **Body orientation (Sec. 3.2), Filter #2.** We project the curated rays parallel to the ground and detect whether any receptacle parts hinder a body from approaching the object from certain directions; the red rays are discarded, while green ones are kept.

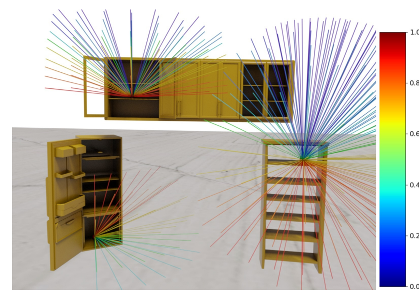

Figure 7. **ReachingField – Ray likelihood (Sec. 3.2, Eq. (1))**, shown with color-coding; red shows high and blue low likelihood. Objects near the ground are likely grasped from above (left). Objects high above the ground are likely grasped from below (right).

Formulation: The *direction* from which a body-arm approaches objects is key for grasping. We provide this to CReach as a normalized vector, $d_{\text{arm}} \in \mathbb{R}^3$. Generated bodies should have an arm direction that aligns with this, so we compute SMPL-X's normalized elbow-to-wrist vector.

Training: We use CIRCLE [3] and GRAB [50] data for training; crucially, the former has a rich range of target wrist locations. We use the direction, d_{arm}, as condition for both encoder $Q(Z|\theta_{wb}, \gamma_{wb}, \beta_{wb}, L_{\text{target}}, d_{\text{arm}})$ and decoder $P(\bar{\theta}_{wb}, \bar{\gamma}_{wb}|Z, \beta_{wb}, L_{\text{target}}, d_{\text{arm}}, h_{\text{int}})$, where Z is the latent code, θ_{wb} is body pose, γ_{wb} is translation, β_{wb} is shape, d_{arm} is the desired arm direction (new over GNet), L_{target} is the target GT wrist joint (as a proxy for object centroid, as CIRCLE has no objects), and h_{int} denotes using the right ($h_{\text{int}} = 0$) or left arm ($h_{\text{int}} = 1$). We add (on top of GNet losses) a loss on arm direction as follows, where $w_{d_{\text{arm}}} = 5$:

$$\mathcal{L}_{d_{\text{arm}}} = w_{d_{\text{arm}}} \cdot \mathbb{E}\left[|d_{\text{arm}} - \bar{d}_{\text{arm}}|\right]. \tag{2}$$

Inference: The decoder takes the arm direction, d_{arm} (from ReachingField), the "target" object centroid, L_{target} (in training we approximate this with the wrist), and parameters β_{wb} and h_{int}, and outputs a SMPL-X body; see Fig. 2.

3.4. CGrasp – Controllable Grasping Hands

Our goal is *controllable* synthesis; we build *CGrasp* by extending GrabNet [50] with a *condition* on *palm direction*.

Formulation: The direction a hand grasps from is key. We provide this to CGrasp as a unit vector, $d_{\text{grasp}} \in \mathbb{R}^3$. All generated hands need to have a palm direction that agrees with d_{grasp}. To this end, we annotate (offline) two vertices on the outer palm of MANO, as it is quasi-rigid so vertices stay consistent during motion. These vertices define d_{grasp}.

Moreover, we enhance the *spatial awareness* of CGrasp. Inspired by GNet [51] and others [16, 61], we compute 3D hand-to-object InterField vectors, $f_{\text{inter}} \in \mathbb{R}^{99\times3}$. In detail, we sample (offline) 99 "interaction" vertices, $v_{h,i}^{\text{inter}}, i \in \{1, \dots, 99\}$, evenly distributed across MANO's inner-palm/finger surface. Then, we compute 3D vectors, f_{inter}, encoding the distance and direction from the sampled hand vertices, v_h^{inter}, to their closest object ones, v'_o.

Training: We train on the GRAB [50] dataset. During training, we add the GT InterField, f_{inter}, to the encoder $Q(Z|BPS_o, f_{\text{inter}})$. In test time, the decoder $P(\bar{\theta}_h, \bar{\gamma}_h, \bar{f}_{\text{inter}}|Z, BPS_o, d_{\text{grasp}})$ predicts MANO parameters, $(\bar{\theta}_h, \bar{\gamma}_h)$, and the InterField, $\bar{f}_{\text{inter}}$. Z is the latent code, and BPS_o is the object shape. We also add (on top of GrabNet losses) a loss on direction and on InterField:

$$\mathcal{L}_{\text{grasp}} = (1 - c_{KL}) \cdot \mathbb{E}\Big[|d_{\text{grasp}} - \bar{d}_{\text{grasp}}|\Big], \quad (3)$$

$$\mathcal{L}_{\text{inter}} = (1 - c_{KL}) \cdot \mathbb{E}\Big[|f_{\text{inter}} - \bar{f}_{\text{inter}}|\Big], \quad (4)$$

where $c_{KL} = 0.005$ is a KL-divergence constant.

Inference: The decoder takes the desired grasp direction, d_{grasp} (sampled from ReachingField), concatenated with the object shape, BPS_o, and outputs a MANO grasp. For inference we append a frozen pretrained RefineNet [50].

3.5. CWGrasp – Whole-Body Synthesis

Given a 3D object lying on a receptacle, we aim to generate a dexterous and physically-plausible SMPL-X body grasp.

Objective function: We build the objective function:

$$\begin{aligned}\mathcal{L}_{\text{opt}} =& \lambda_{hm}\mathcal{L}_{hm} + \lambda_\theta\mathcal{L}_\theta + \lambda_g\mathcal{L}_g + \\ & \lambda_{grd}\mathcal{L}_{grd} + \lambda_p\mathcal{L}_p + \lambda_{reg}\mathcal{L}_{reg},\end{aligned} \quad (5)$$

consisting of a hand-matching term, $\mathcal{L}_{hm}$, a body pose term, $\mathcal{L}_\theta$, a head-direction term, $\mathcal{L}_g$ (often called "gaze"), a ground-body penetration term, $\mathcal{L}_{grd}$, a receptacle-body penetration term, $\mathcal{L}_p$, and a regularizer term, $\mathcal{L}_{reg}$. These terms are similar to FLEX [54], except for $\mathcal{L}_{grd}$ and $\mathcal{L}_{reg}$. For details on our loss terms, see Sup. Mat. (Sec. S.1.4).

Search space: We operate in the original search space [26, 44] for flexibility. This contrasts to FLEX [54] that uses a compact "black-box" latent space but loses some control. Even if CReach generates a body from a desired approaching direction, sometimes the body penetrates the receptacle (see Fig. 9-left). Starting the optimization from such a local minimum might trap the optimizer. To prevent this, we first translate the body by 1m along the the floor-projected direction used to condition CReach, so we free it from big penetrations (see Fig. 9-middle). Then, optimization (Eq. (5)) pulls the body back to the object while refining body and finger pose (see Fig. 9-right). This makes CWGrasp robust.

Optimizer: We use Adam; for 1 body and for 1500 iterations it takes $\sim$ 20 sec on an Nvidia RTX 4500-Ada GPU.

Sample efficiency: We sample from ReachingField just one direction and condition on it both CReach and CGrasp. Thus, our reaching body and guiding hand are already compatible, and refine only the body to match the (fixed) hand. Instead, FLEX [54] samples 500 bodies, and refines both bodies and guiding hands, due to using the non-controllable GrabNet. Therefore, our method is very sample efficient.

Left-hand interaction for whole bodies: CWGrasp uniquely generates both right- and left-hand whole-body grasps. For the latter, conditioning CReach with $h_{\text{int}} = 1$ (see Sec. 3.3) produces a body that reaches the object with its left arm. Then, we mirror both the object and ReachingField's direction (w.r.t. the object's sagittal plane), generate a right hand grasp with CReach, and mirror back the hand and object. Last, we run CWGrasp optimization.

4. Experiments

4.1. Conditioning for CReach & CGrasp

We evaluate how accurately CGrasp and CReach preserve their conditioning, i.e., the desired arm and palm direction. Table 1 reports results (incl. runtime) computed as follows.

CGrasp: This is conditioned on a palm direction vector. We extract all hand directions from GRAB's [50] test set and cluster them into 200 centers using K-Means. We then use GRAB's 6 test objects and generate for each of these 2000 grasps; to this end, we run CGrasp 10 times per cluster center while conditioning on its direction. We then compute the palm direction of each generated grasp and its angular error w.r.t. the conditioning direction. A mean angular error of 4.57°denotes accurate generation; this is also reflected in qualitative results in Sup. Mat. (Fig. S.2).

CReach: This is conditioned on an arm direction and wrist location. We extract all arm directions and wrist locations of ReplicaGrasp's [54] and GRAB's [50] test sets, and cluster each of these 2 modalities into 200 centers via K-means. With these, we obtain 40000 combinations of arm directions and wrist locations for conditioning CReach and generating 40000 reaching bodies for each (left/right) arm. Then, we compute over all generated bodies the mean angular error for arm direction (as above for palm direction for CGrasp), and the Mean Squared Error (MSE) for wrist locations. The values in Tab. 1 denote accurate synthesis.

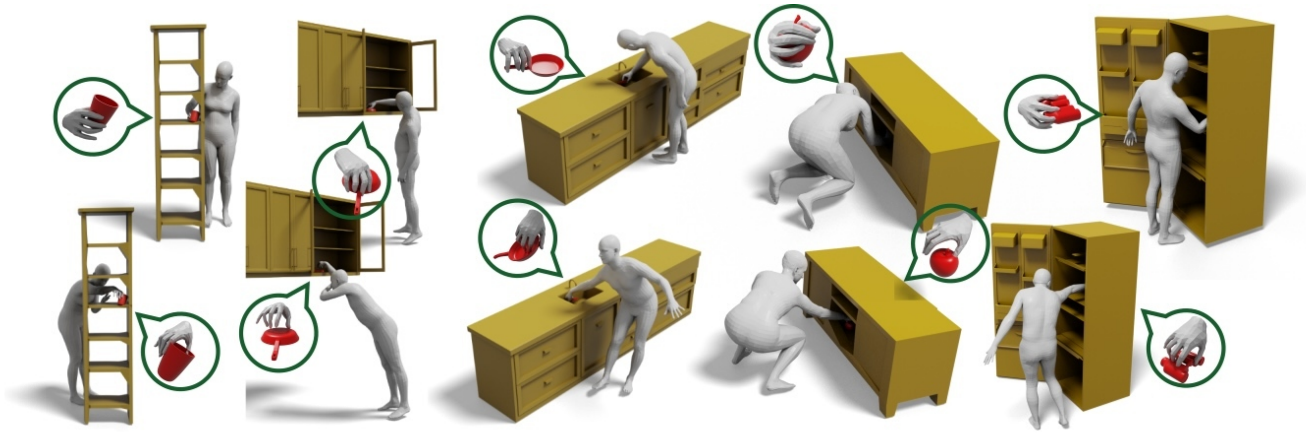

Figure 8. **Whole-body grasps** produced by CWGrasp (top row) and FLEX [54] (bottom). FLEX samples 500 initial bodies and produces 10 ones; we show the smallest-loss one. Our CWGrasp samples only 1 body and also generates one, yet it produces more realistic grasps.

4.2. Hand-Only Grasps (CGrasp)

We evaluate CGrasp on the GRAB [50] dataset against DexGraspNet [58], ContactGen [36], and GrabNet [50]. For each method, we generate 200 grasps for each of the 6 test objects, and compute the following metrics as in [28, 29, 31, 36, 56, 59, 64, 66]. We report results in Tab. 2

Contact ratio [36]: We detect the contacting MANO vertices by thresholding its distances (1mm) from the object, and compute the ratio of these over all MANO vertices.

Penetration percentage (%) [54]: We compute the percentage of hand vertices penetrating the object via the signed distances of the two meshes (distance ≤ -1mm).

Penetration volume [28]: We voxelize the hand and object meshes using voxels of volume $v = 1\text{mm}^3$, and detect intersecting voxels N. Then, the penetration volume is $N \cdot v$.

Penetration depth [28]: We compute the minimum translation along the opposite palmar direction (d_{grasp} in Sec. 3.4) necessary for resolving any hand-object penetrations.

Hand pose diversity [54]: We align all hands at the same wrist location and palm orientation, and compute the mean Euclidean vertex distance over all possible mesh pairs.

Table 2 shows that CGrasp performs on par with baselines. That is, CGrasp's benefit of controllability does not harm performance. We show qualitative results in Sup. Mat. (Fig. S.4); these reflect quantitative ones. We also compare contact heatmaps in Fig. 10. Baselines involve mainly the fingertips, while CGrasp involves also parts of the palm.

4.3. Whole-Body Grasps (CWGrasp)

We evaluate CWGrasp on the ReplicaGrasp dataset [54] and compare it against the state-of-the-art FLEX [54] method.

Experimental setup: ReplicaGrasp places GRAB [50] objects on various receptacles, e.g., sofas, tables. Each of the 50 GRAB objects appears in 192 configurations, varying the receptacle and the object's location and orientation

	Angle (degrees) ↓	MSE (cm) ↓	Inf. time (s) ↓
CReach-RA	7.67	4	0.46
CReach-LA	7.23	3.6	0.46
CGrasp	4.57	N/A	0.47

Table 1. **Condition accuracy.** CReach and CGrasp generate bodies and hands conditioned on a (arm/hand) direction. We report the angular error of the arm/palm direction, the Mean Squared Error (MSE) of wrist joints, and inference time. For CReach we evaluate right- (RA) and left-arm (LA) reaching.

Figure 9. **CReach failure.** CReach might produce a reaching body that penetrates the receptacle (left). To correct for this, we translate the body by 1m (middle) along the opposite floor-projected arm direction. Then, CWGrasp's optimization (Sec. 3.5) pulls the body back to the object, while refining body and finger pose (right).

on it. For our experiments, we use the 6 test objects and 6 randomly-sampled training objects of GRAB, and randomly select 20 configurations per object. For each configuration, we generate grasping bodies with both CWGrasp and FLEX and compare the two methods. Note that FLEX optimizes 500 samples, and eventually keeps 10 samples with smaller losses; we consider the "best" (smallest-loss) one. Instead, our CWGrasp uses only a single sample.

Quantitative evaluation: We report the five metrics defined in Sec. 4.2 also here in Tab. 3, but with the following adaptations due to switching to whole-body context. We compute the "penetration percentage" separately for body–

receptacle ($\mathcal{B}-\mathcal{M}$) and for right-hand–object ($\mathcal{RH}-\mathcal{O}$) interaction. We compute the "contact ratio" for right-hand–object ($\mathcal{RH}-\mathcal{O}$) interaction. We compute the "body pose diversity" by extending "hand pose diversity" to whole-body meshes. Last, we report the mean optimization time for each method. We observe that our CWGrasp framework is highly competitive against FLEX, while using 500$\times$ less samples and being one order of magnitude faster.

Qualitative evaluation: We visualize several whole-body grasps produced by CWGrasp and FLEX in Fig. 8. We observe that CWGrasp produces more natural-looking body poses. For many more qualitative results, including close-up views into hands, as well as left-hand whole-body interactions, please see Sup. Mat. (Sec. S.3). We also compare aggregated contact heatmaps in Fig. 11. We see that FLEX grasps tend to use mainly the fingertips, while CWGrasp grasps activate also parts of the palm, so they look richer.

Perceptual Study: To evaluate the perceived realism of generated grasps, we conduct a perceptual study. To this end, we sample object-and-receptacle configurations from the ReplicaGrasp [54] dataset, and for each one, we generate two whole-body grasps (referred to as "samples") with CWGrasp and FLEX, respectively. For each sample, we conduct two comparisons by rendering a whole-body view and a zoomed-in view onto the hand and object (see examples in Sup. Mat. Fig. S.8). We randomize the order that we present samples, as well as their placement. Each sample is shown to 35 participants, who choose which method generates the most realistic grasp (see the protocol shown to participants in Sup. Mat. Fig. S.7). In total, we show 28 samples, of which 4 are catch trials (letting us filter out 2 of the participants). Considering the full-body view, CWGrasp is preferred 70.8% of the times. Considering the zoomed-in view, it is preferred 71.6% of the times. Considering both views, it is preferred 71.23% of the times. That is, our CWGrasp produces grasps that are perceived as significantly more realistic than the state of the art.

5. Conclusion

We develop CWGrasp, a method that generates whole-body grasps for objects through novel *controllable synthesis*. To this end, we first learn ReachingField, a novel model for estimating directions a body can approach the object from. However, current body and grasp generators lack controllability. To fill this gap, we learn the novel CReach and CGrasp models that generate a reaching body and a grasping hand with a desired arm and palm 3D direction, respectively. We condition both CReach and CGrasp on the same direction sampled from ReachingField to produce a grasping hand and reaching body that are compatible with each other. Last, our CWGrasp method combines these with only a small refinement, efficiently producing grasps that are perceived as significantly more realistic than the state of the art.

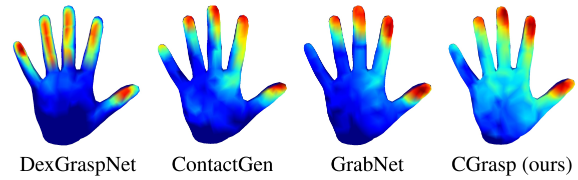

Figure 10. **Contact maps: CGrasp & SotA (Sec. 4.2).** Contact likelihood is color-coded via heatmaps; red denotes a high likelihood and blue a low one. We compare against DexGraspNet [58], ContactGen [36], and GrabNet [50]. Existing methods involve mostly finger tips, while CGrasp also involves parts of the palm.

	Type	**Control**	**Cont. ratio** $\uparrow$	**Penetr. perc.** $\% \downarrow$	**Penetr. vol.** $\downarrow$ mm^3	**Penetr. depth** $mm \downarrow$	**Hand pose div.** $cm \uparrow$
DexGraspNet	O	✗	0.11	**0.13**	1.25	**1.2**	**7.08**
ContactGen	R	✗	0.09	1.15	**1.04**	2	6.75
GrabNet	R	✗	**0.13**	2.4	1.27	2.6	6.72
CGrasp (ours)	R	✓	0.12	2.9	1.16	2.8	6.72

Table 2. **Evaluation: CGrasp & SotA (Sec. 4.2).** The "type" column denotes regression (R) or optimization (O) methods. The "control" column indicates whether a method is controllable via directional conditioning. Our CGrasp performs *on par* with existing methods, while being controllable via a direction condition. That is, the benefit of *controllability does not harm* performance.

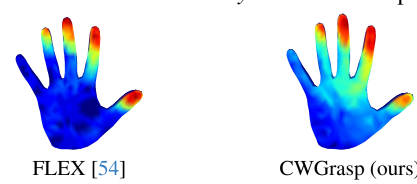

Figure 11. **Contact maps: CWGrasp & FLEX [54] (Sec. 4.3).** Contact likelihood is color-coded via heatmaps; red denotes a high likelihood and blue a low one. FLEX involves mainly the fingertips, while our CWGrasp also involves parts of the pam.

	Samples #	*Pen.* % $\mathcal{B}-\mathcal{M}$ $\downarrow$	*Pen.* % $\mathcal{RH}-\mathcal{O}$ $\downarrow$	*Contact* $\mathcal{RH}-\mathcal{O}$ $\uparrow$	Body div. $cm \uparrow$	Time (s) $\downarrow$
FLEX [54]	500	**0.3**	1.16	0.15	**63.86**	357
CWGrasp	1	0.7	**0.7**	**0.3**	61.77	**23**

Table 3. **Evaluation: CWGrasp & FLEX (Sec. 4.3):** We report the number of body samples, the number of optimization iterations, the penetration percentage for the whole body ($\mathcal{B}$) and receptacle ($\mathcal{M}$), and for the right-hand ($\mathcal{RH}$) and object ($\mathcal{O}$), the contact ratio, body pose diversity, and average runtime.

Future Work: We tackle right- and left-hand grasps; future work will look into bi-manual grasping [16, 22, 50, 61]. Sometimes bodies look "unstable" when kneeling down or stretching up; intuitive-physics reasoning [55] might help. Last, we will use generated grasps as targets for motion synthesis [27, 51, 57, 59] to navigate scenes and grasp objects.

Acknowledgments: This work is partially supported by the ERC Starting Grant (project STRIPES, 101165317, PI: D. Tzionas).
Disclosure: D. Tzionas has received a research gift from Google.

References

[1] Brett Allen, Brian Curless, Zoran Popovic, and Aaron Hertzmann. Learning a correlated model of identity and pose-dependent body shape variation for real-time synthesis. In *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 147–156, 2006. 3

[2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: Shape completion and animation of people. In *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, page 408–416, 2005. 3

[3] João Pedro Araújo, Jiaman Li, Karthik Vetrivel, Rishi Agarwal, Jiajun Wu, Deepak Gopinath, Alexander Clegg, and C. Karen Liu. CIRCLE: Capture in rich contextual environments. In *Computer Vision and Pattern Recognition (CVPR)*, pages 21211–21221, 2023. 2, 3, 5, 12

[4] Luca Ballan, Aparna Taneja, Jürgen Gall, Luc Van Gool, and Marc Pollefeys. Motion capture of hands in action using discriminative salient points. In *European Conference on Computer Vision (ECCV)*, pages 640–653, 2012. 3

[5] Antonio Bicchi and Vijay Kumar. Robotic grasping and contact: A review. In *International Conference on Robotics and Automation (ICRA)*, pages 348–353, 2000. 3

[6] Samarth Brahmbhatt, Cusuh Ham, Charles C. Kemp, and James Hays. ContactDB: Analyzing and predicting grasp contact via thermal imaging. In *Computer Vision and Pattern Recognition (CVPR)*, pages 8709–8719, 2019. 3

[7] Samarth Brahmbhatt, Ankur Handa, James Hays, and Dieter Fox. ContactGrasp: Functional multi-finger grasp synthesis from contact. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 2386–2393, 2019. 3

[8] Samarth Brahmbhatt, 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

[9] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S. Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, Jan Kautz, and Dieter Fox. DexYCB: Abenchmark for capturing hand grasping of objects. In *Computer Vision and Pattern Recognition (CVPR)*, pages 9044–9053, 2021. 3

[10] Sammy Joe 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)*, pages 20577–20586, 2022. 3

[11] Enric Corona, Albert Pumarola, Guillem Alenyà, Francesc Moreno-Noguer, and Grégory Rogez. GanHand: Predicting human grasp affordances in multi-object scenes. In *Computer Vision and Pattern Recognition (CVPR)*, pages 5031–5041, 2020. 3

[12] Enric Corona, Tomas Hodan, Minh Vo, Francesc Moreno-Noguer, Chris Sweeney, Richard A. Newcombe, and Lingni Ma. LISA: learning implicit shape and appearance of hands. In *Computer Vision and Pattern Recognition (CVPR)*, pages 20501–20511, 2022. 3

[13] Mark R. Cutkosky. On Grasp Choice, Grasp Models, and the Design of Hands for Manufacturing Tasks. *Transactions on Robotics and Automation (TRA)*, 5(3):269–279, 1989. 3

[14] Mark R. Cutkosky and Paul K. Wright. Modeling manufacturing grips and correlations with the design of robotic hands. In *International Conference on Robotics and Automation (ICRA)*, pages 1533–1539, 1986. 3

[15] Markos Diomataris, Nikos Athanasiou, Omid Taheri, Xi Wang, Otmar Hilliges, and Michael J. Black. WANDR: Intention-guided human motion generation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 927–936, 2024. 3

[16] 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, 6, 8

[17] 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 *European Conference on Computer Vision (ECCV)*, pages 428–448, 2024. 3

[18] Zicong Fan, Maria Parelli, Maria Eleni Kadoglou, Xu Chen, Muhammed Kocabas, Michael J. Black, and Otmar Hilliges. HOLD: Category-agnostic 3D reconstruction of interacting hands and objects from video. In *Computer Vision and Pattern Recognition (CVPR)*, pages 494–504, 2024. 3

[19] Thomas Feix, Javier Romero, Heinz-Bodo Schmiedmayer, Aaron M. Dollar, and Danica Kragic. The GRASP taxonomy of human grasp types. *Transactions on Human-Machine Systems (THMS)*, 46(1):66–77, 2016. 3

[20] 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)*, pages 409–419, 2018. 3

[21] Dariu Gavrila and Larry S. Davis. 3-D Model-based tracking of humans in action: A multi-view approach. In *Computer Vision and Pattern Recognition (CVPR)*, pages 73–80, 1996. 3

[22] Anindita Ghosh, Rishabh Dabral, Vladislav Golyanik, Christian Theobalt, and Philipp Slusallek. IMoS: Intent-driven full-body motion synthesis for human-object interactions. In *Computer Graphics Forum (CGF)*, 2023. 3, 8

[23] Patrick Grady, Chengcheng Tang, Christopher D. Twigg, Minh Vo, Samarth Brahmbhatt, and Charles C. Kemp. ContactOpt: Optimizing contact to improve grasps. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1471–1481, 2021. 3

[24] 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 3196–3206, 2020. 3

[25] 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 *Computer Vision and Pattern Recognition (CVPR)*, pages 11090–11100, 2022. 3

[26] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J. Black. Resolving 3D human pose ambiguities with 3D scene constraints. In *International Conference on Computer Vision (ICCV)*, pages 2282–2292, 2019. 2, 6

[27] 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, 8

[28] Yana Hasson, Gül Varol, Dimitrios 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, 7

[29] 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, 7

[30] 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. 3

[31] 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. 7

[32] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J. Black, Krikamol Muandet, and Siyu Tang. Robotic grasping from classical to modern: A survey. *arXiv:2202.03631*, 2022. 3

[33] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2O: Two hands manipulating objects for first person interaction recognition. In *International Conference on Computer Vision (ICCV)*, pages 10138–10148, 2021. 3

[34] Jiaman Li, Jiajun Wu, and C. Karen Liu. Object motion guided human motion synthesis. In *Transactions on Graphics (TOG)*, pages 197:1–197:11, 2023. 3

[35] Yuwei Li, Longwen Zhang, Zesong Qiu, Yingwenqi Jiang, Nianyi Li, Yuexin Ma, Yuyao Zhang, Lan Xu, and Jingyi Yu. NIMBLE: A non-rigid hand model with bones and muscles. *Transactions on Graphics (TOG)*, 41(4):120:1–120:16, 2022. 3

[36] Shaowei Liu, Yang Zhou, Jimei Yang, Saurabh Gupta, and Shenlong Wang. ContactGen: Generative contact modeling for grasp generation. In *International Conference on Computer Vision (ICCV)*, pages 20552–20563, 2023. 3, 7, 8, 14

[37] Xueyi Liu and Li Yi. GeneOH diffusion: Towards generalizable hand-object interaction denoising via denoising diffusion. In *International Conference on Learning Representations (ICLR)*, 2024. 3

[38] 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 *Computer Vision and Pattern Recognition (CVPR)*, pages 21013–21022, 2022. 3

[39] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *Transactions on Graphics (TOG)*, 34(6):1–16, 2015. 3

[40] D. Marr and H. K. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 200(1140):269–294, 1978. 3

[41] Andrew T. Miller, Steffen Knoop, Henrik I. Christensen, and Peter K. Allen. Automatic grasp planning using shape primitives. In *International Conference on Robotics and Automation (ICRA)*, pages 1824–1829, 2003. 3

[42] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A. Argyros. Efficient model-based 3D tracking of hand articulations using Kinect. In *British Machine Vision Conference (BMVC)*, pages 1–11, 2011. 3

[43] Ahmed A. A. Osman, Timo Bolkart, Dimitrios Tzionas, and Michael J. Black. SUPR: A sparse unified part-based human body model. In *European Conference on Computer Vision (ECCV)*, pages 568–585, 2022. 3

[44] 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. 2, 3, 4, 6

[45] Sergey Prokudin, Christoph Lassner, and Javier Romero. Efficient Learning on Point Clouds With Basis Point Sets. In *International Conference on Computer Vision (ICCV)*, pages 4332–4341, 2019. 3, 4

[46] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *Transactions on Graphics (TOG)*, 36(6):1–17, 2017. 2, 3, 4

[47] Srinath Sridhar, Antti Oulasvirta, and Christian Theobalt. Interactive markerless articulated hand motion tracking using rgb and depth data. In *International Conference on Computer Vision (ICCV)*, pages 2456–2463, 2013. 3

[48] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *Transactions on Graphics (TOG)*, 38(6):1–14, 2019. 3

[49] Anilkumar Swamy, Vincent Leroy, Philippe Weinzaepfel, Fabien Baradel, Salma Galaaoui, Romain Brégier, Matthieu Armando, Jean-Sébastien Franco, and Grégory Rogez. SHOWMe: Benchmarking object-agnostic hand-object 3D reconstruction. In *International Conference on Computer Vision (ICCV)*, pages 1927–1936, 2023. 3

[50] 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. 2, 3, 4, 5, 6, 7, 8, 12, 14

[51] 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 13263–13273, 2022. 2, 3, 4, 5, 6, 8, 14

[52] Omid Taheri, Yi Zhou, Dimitrios Tzionas, Yang Zhou, Duygu Ceylan, Soren Pirk, and Michael J. Black. GRIP: Generating interaction poses using spatial cues and latent consistency. In *International Conference on 3D Vision (3DV)*, pages 933–943, 2024. 3

[53] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+O: Unified egocentric recognition of 3D hand-object poses and interactions. In *Computer Vision and Pattern Recognition (CVPR)*, pages 4511–4520, 2019. 3

[54] Purva Tendulkar, Dídac Surís, and Carl Vondrick. FLEX: Full-body grasping without full-body grasps. In *Computer Vision and Pattern Recognition (CVPR)*, pages 21179–21189, 2023. 1, 2, 3, 6, 7, 8, 14, 15, 16, 17, 18, 19

[55] Shashank Tripathi, Lea Müller, Chun-Hao P. Huang, Taheri Omid, Michael J. Black, and Dimitrios Tzionas. 3D human pose estimation via intuitive physics. In *Computer Vision and Pattern Recognition (CVPR)*, pages 4713–4725, 2023. 8, 14

[56] Dylan Turpin, Liquang Wang, Eric Heiden, Yun-Chun Chen, Miles Macklin, Stavros Tsogkas, Sven J. Dickinson, and Animesh Garg. Grasp'D: Differentiable contact-rich grasp synthesis for multi-fingered hands. In *European Conference on Computer Vision (ECCV)*, pages 201–221, 2022. 3, 7

[57] 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, 8

[58] Ruicheng Wang, Jialiang Zhang, Jiayi Chen, Yinzhen Xu, Puhao Li, Tengyu Liu, and He Wang. DexGraspNet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. In *International Conference on Robotics and Automation (ICRA)*, pages 11359–11366, 2022. 3, 7, 8, 14

[59] Yan Wu, Jiahao Wang, Yan Zhang, Siwei Zhang, Otmar Hilliges, Fisher Yu, and Siyu Tang. SAGA: Stochastic whole-body grasping with contact. In *European Conference on Computer Vision (ECCV)*, pages 257–274, 2022. 3, 7, 8

[60] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T. Freeman, Rahul Sukthankar, and Cristian Sminchisescu. GHUM & GHUML: Generative 3D human shape and articulated pose models. In *Computer Vision and Pattern Recognition (CVPR)*, pages 6183–6192, 2020. 3

[61] 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. 3, 6, 8

[62] Hui Zhang, Sammy Christen, Zicong Fan, Otmar Hilliges, and Jie Song. GraspXL: Generating grasping motions for diverse objects at scale. In *European Conference on Computer Vision (ECCV)*, pages 386–403, 2024. 3

[63] Mengqi Zhang, Yang Fu, Zheng Ding, Sifei Liu, Zhuowen Tu, and Xiaolong Wang. HOIDiffusion: Generating realistic 3D hand-object interaction data. In *Computer Vision and Pattern Recognition (CVPR)*, pages 8521–8531, 2024. 3

[64] Siwei Zhang, Yan Zhang, Qianli Ma, Michael J. Black, and Siyu Tang. PLACE: Proximity learning of articulation and contact in 3D environments. In *International Conference on 3D Vision (3DV)*, pages 642–651, 2020. 7

[65] Xiaohan Zhang, Bharat Lal Bhatnagar, Sebastian Starke, Vladimir Guzov, and Gerard Pons-Moll. COUCH: Towards controllable human-chair interactions. In *European Conference on Computer Vision (ECCV)*, pages 518–535, 2022. 3

[66] 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, 2019. 7

[67] Keyang Zhou, Bharat Lal Bhatnagar, Jan Eric Lenssen, and Gerard Pons-Moll. TOCH: Spatio-temporal object-to-hand correspondence for motion refinement. In *European Conference on Computer Vision (ECCV)*, pages 1–19, 2022. 3

[68] Keyang Zhou, Bharat Lal Bhatnagar, Jan Eric Lenssen, and Gerard Pons-Moll. GEARS: Local geometry-aware hand-object interaction synthesis. In *Computer Vision and Pattern Recognition (CVPR)*, pages 20634–20643, 2024. 3

[69] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan C. Russell, Max J. Argus, and Thomas Brox. FreiHAND: A Dataset for Markerless Capture of Hand Pose and Shape From Single RGB Images. In *International Conference on Computer Vision (ICCV)*, pages 813–822, 2019. 3