3D Whole-Body Grasp Synthesis with Directional Controllability

Supplementary Material

S.1. Implementation Details

Here we describe details of our methodology. Section S.1.1 discusses details on the steps for building the ReachingField probabilistic model. Section S.1.2 discusses details for the CReach model. Section S.1.3 discusses details for the CGrasp model. Section S.1.4 discusses details of our CWGrasp framework that employs all above components.

S.1.1. ReachingField

We discuss details for building a ReachingField, and visualize steps in Fig. S.1, where we show examples for varying object "heights," namely distances from the ground.

Given an object (see Fig. S.1 A), we first define a spherical grid around the object (see Fig. S.1 B), and cast rays towards all directions formed between the object centroid and the spherical-grid points (see Fig. S.1 C). We then follow a filtering process that includes the following steps:

Filter #1. Arm/hand direction: We only keep the casted rays that do not intersect with the receptacle; see Fig. S.1 D.

Filter #2 - Body orientation: We project the remaining rays to the horizontal plane and check again for intersections with the receptacle; see Fig. S.1 E.

Filter #3 - Standing places: We sample points across the remaining rays and cast vertical rays towards the ground (see Fig. S.1 F). Then, we detect potential "occluders" that lie beneath a ray and prevent a human from standing there and approaching the object along that ray direction.

Filter #4 - Wiggle room for arm volume: Some remaining rays might still result in body-receptacle penetrations, as ray casting does not consider the volume of body limbs. To account for this we "perturb" ray directions while checking for intersections, to allow for a "wiggle room" that can be occupied by a body's arm. To speed up this process, we empirically observe that we tend to approach an object from the right to interact with the right hand, and from the left to interact with the left hand. Thus, we rotate rays clockwise around the vertical axis (rotating arbitrarily could generalize better but would be slower) within a range of $[0^o, 30^o]$ for right-hand interactions, and counterclockwise for left-hand ones, while pruning intersecting rays. For a visualization see the video on our website. The above is a simple speed-up heuristic that empirically works well for our scenarios.

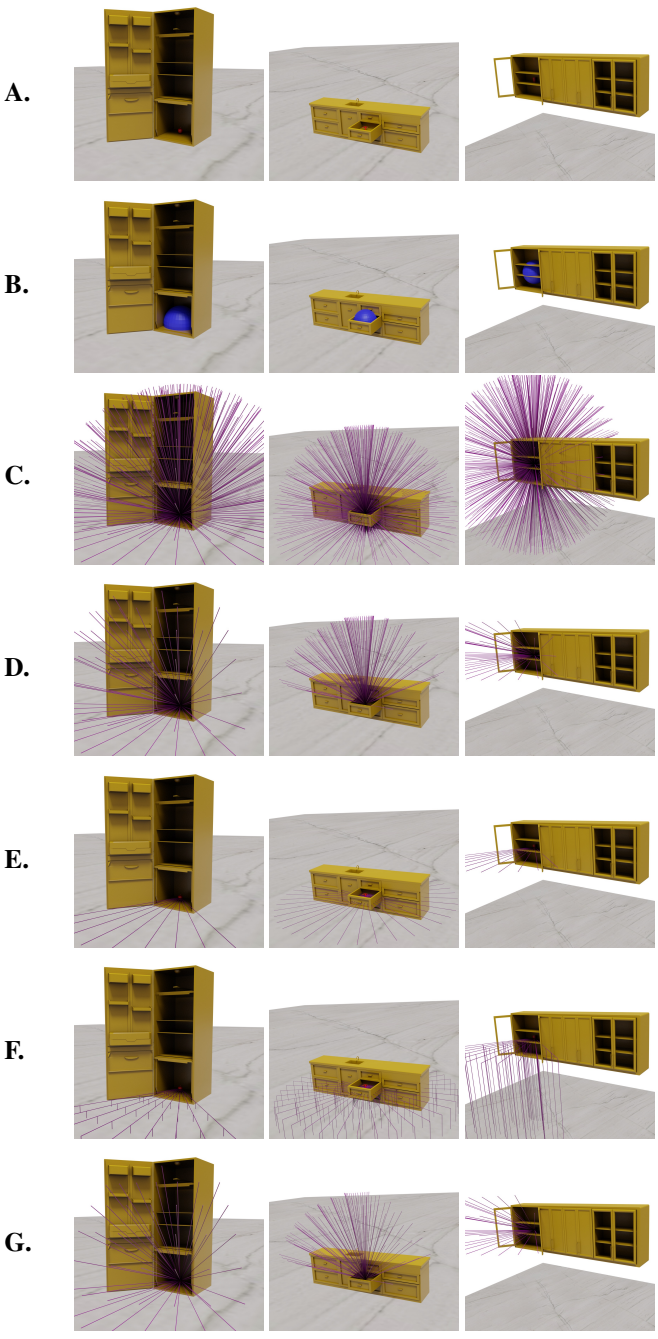

Figure S.1. ReachingField steps for three object "heights."

S.1.2. CReach

Training datasets: To train our CReach model we combine the GRAB [50] and CIRCLE [3] datasets. GRAB captures dexterous interactions, but is limited mainly to standing bodies. Instead, CIRCLE captures bodies without fingers, but has a wide range of "reaching" body poses, including kneeling down and stretching up. To get the best of both worlds, namely rich body poses with dexterous fingers, we combine GRAB and CIRCLE.

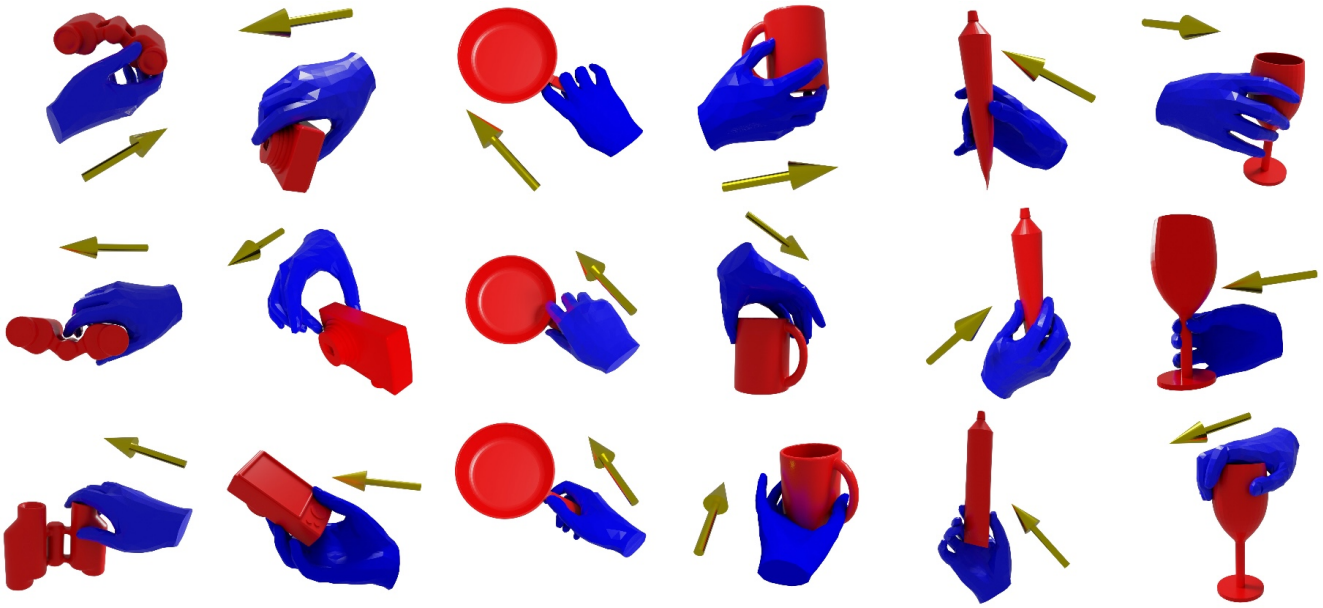

Figure S.2. Qualitative results of CGrasp, along with the given direction vectors. For the six test objects in the GRAB dataset, we generate three grasps per object using different direction vectors as conditions. The conditioning direction vectors are shown by gold arrows.

Note that these datasets capture both left- and right-arm interactions. To train a single network for these, we need to identify the interaction "handedness" in the data, since relevant annotations are missing. We achieve this by exploiting the hand-eye coordination taking place for grasping. To this end, for each body, we compute three vectors: the gaze vector, g_{dir}, via two pre-annotated vertices at the nose tip and the back of the head, and two vectors, d_{erw} and d_{elw}, from the glabella (point between the two eyes) to the right and left wrist joints, respectively. We will release the annotated vertex IDs. Then, interaction "handedness" is defined as:

$$\begin{aligned} \widehat{g_{dir}d_{erw}} \leq \widehat{g_{dir}d_{elw}} &: \textbf{right-hand interaction} \\ \widehat{g_{dir}d_{erw}} > \widehat{g_{dir}d_{elw}} &: \textbf{left-hand interaction} \end{aligned} \tag{S.1}$$

Since right-hand interactions are much more frequent than left-hand ones in GRAB and CIRCLE, we mirror all data for balancing the "handedness."

S.1.3. CGrasp

CGrasp is a CVAE that exploits an InterField and a vector denoting the desired palm direction. Specifically, the InterField is input to the encoder, while the direction vector conditions the decoder. Figure S.2 shows grasps generated by our CGrasp for the 6 test objects of the GRAB dataset, along with the corresponding direction vectors used as condition. In all cases, the direction of the generated grasp "agrees" with the conditioning direction vector. Figure S.3 shows a visual example of the InterField. Figure S.4 shows qualitative results of CGrasp compared to existing methods.

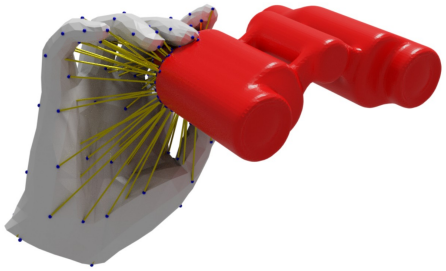

Figure S.3. Visual representation of the InterField. We depict with blue spheres the 99 sampled "interaction" hand vertices, $v_{h,i}^{\text{inter}}$, $i \in \{1, \ldots, 99\}$, evenly-distributed across MANO's inner palm/finger surface. With olive-color lines we depict the 3D vectors, f_{inter}, that encode the distance and direction from the sampled hand vertices, v_h^{inter}, to their closest object vertices, v_o'.

S.1.4. CWGrasp – Optimization details

Our goal is to optimize over the SMPL-X pose, translation, and global-orientation parameters, so that the hand of the body aligns with the "guiding" hand generated by CGrasp, while contacting the ground without penetrating the receptacle. To this end, we use the objective function of Eq. (5):

$$\begin{aligned} \mathcal{L}_{opt} = &\lambda_p \mathcal{L}_p + \lambda_{grd} \mathcal{L}_{grd} + \lambda_\theta \mathcal{L}_\theta + \\ &\lambda_g \mathcal{L}_g + \lambda_{hm} \mathcal{L}_{hm} + \lambda_{reg} \mathcal{L}_{reg}. \end{aligned}$$

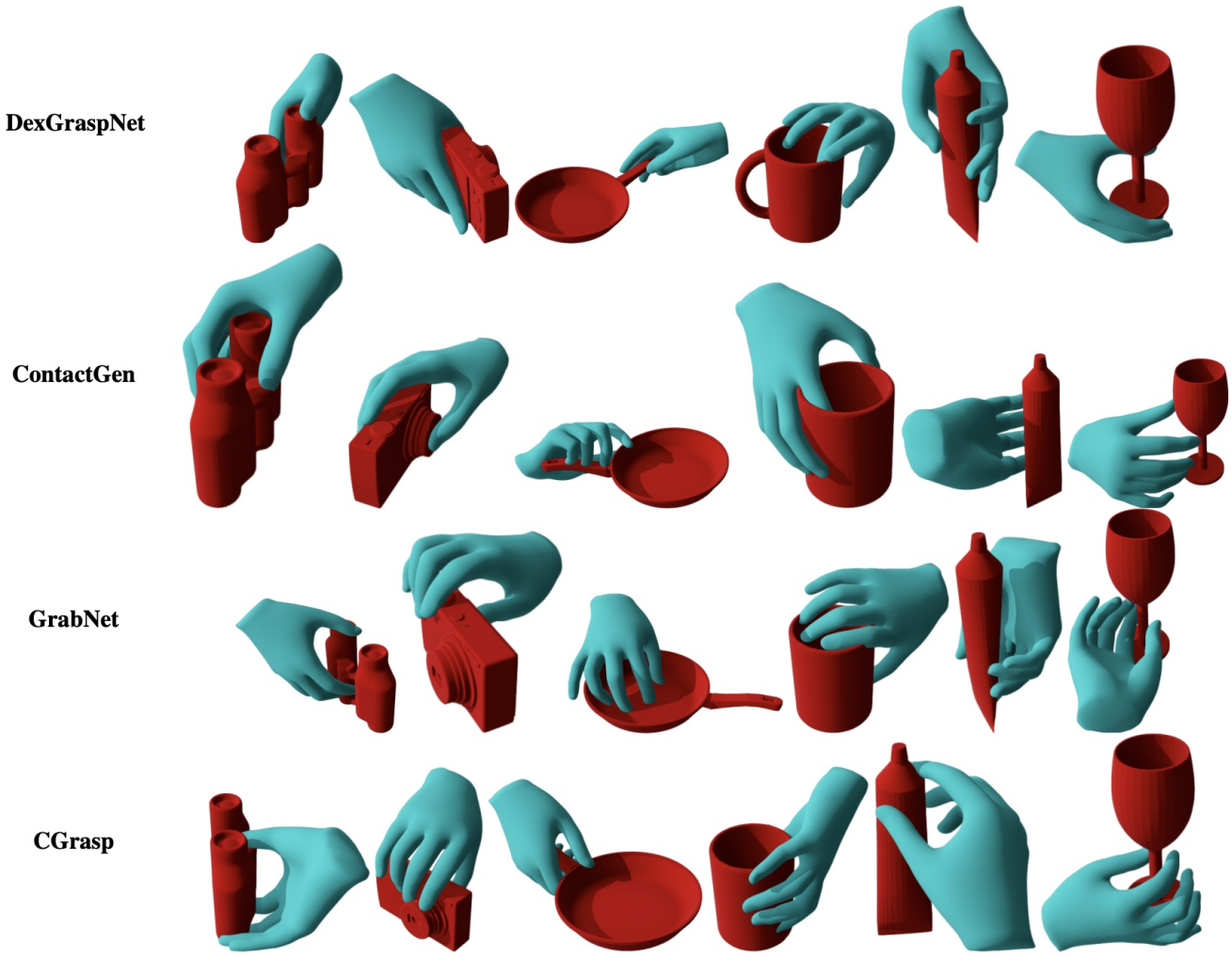

Figure S.4. Comparison of hand-only grasp synthesis models: DexGraspNet [58], ContactGen [36], GrabNet [50], and CGrasp (ours). Each row shows grasps generated by a different model for the same object set (binoculars, camera, frying pan, mug, toothpaste, wineglass).

The term $\mathcal{L}_p$ is a penetration loss between the body and the receptacle, consisting of two terms:

$$\begin{aligned} \mathcal{L}_p &= \mathcal{L}_{p_{inter}} + \mathcal{L}_{p_{con}}, \\ \mathcal{L}_{p_{inter}} &= \frac{1}{N_{\mathcal{V}_b}} \sum_{i=1}^{N_{\mathcal{V}_b}} \mid min(0, d(\mathcal{V}_{b_i}, \mathcal{M})) \mid , \\ \mathcal{L}_{p_{con}} &= \frac{1}{N_{\mathcal{V}_b}} \sum_{i=1}^{N_{\mathcal{V}_b}^{disc}} d(\mathcal{V}_{b_i}^{disc}, \mathcal{M})), \end{aligned} \tag{S.2}$$

where $d(\cdot)$ denotes the signed distance function between the vertices of the body, $\mathcal{V}_b$, and the receptacle mesh, $\mathcal{M}$. The first term penalizes vertices that penetrate the receptacle $\mathcal{M}$. The second term penalizes the body vertices that get "disconnected" from the rest of the body when the latter gets "intercepted" by the penetrated receptacle [54].

For the ground loss, $\mathcal{L}_{grd}$, we first find the height, $h(\mathcal{V}_{b_i})$, of body vertices w.r.t. the ground plane. Then we have two cases. For vertices that penetrate the ground we use the under-ground term of [55]:

$$\mathcal{L}_{grd} = \beta_1 tanh \left(\frac{h\left(\mathcal{V}_{b_i}\right)}{\beta_2} \right)^2, \quad \text{for } h(\mathcal{V}_{b_i}) < 0, \tag{S.3}$$

where $\beta_1 = 1$ and $\beta_2 = 0.15$, as in [55]. For vertices above the ground (i.e., for missing contacts), we use:

$$\mathcal{L}_{grd} = \mid min(\mathcal{V}_{b_i z}) \mid , \quad \text{for } h(\mathcal{V}_{b_i}) > 0, \tag{S.4}$$

where $i = 1 \cdots N_{\mathcal{V}_b}$. Both Eq. (S.3) and Eq. (S.4) are responsible for keeping bodies on the ground.

To ensure that the generated body has "eye contact" with the object of interaction we use the gaze loss of GOAL [51]. To this end, we annotate two vertices $\mathcal{A}$ and $\mathcal{B}$ on the head,

namely vertex $\mathcal{A}$ at the back of the head, and vertex $\mathcal{B}$ lying between the eyes. The 3D position of the object $\mathcal{O}$ is known. Thus, we define the vectors $\overrightarrow{\mathcal{AO}}$ and $\overrightarrow{\mathcal{BO}}$ and specify the gaze loss as their in-between angle:

$$\mathcal{L}_g = cos^{-1}\left(\frac{\overrightarrow{\mathcal{AO}} \quad \overrightarrow{\mathcal{BO}}}{|\overrightarrow{\mathcal{AO}}||\overrightarrow{\mathcal{BO}}|}\right). \tag{S.5}$$

As we optimize over the SMPL-X pose parameters, we need to encourage the predicted poses, $\hat{\theta}$, to remain on the manifold of our CReach, so that we produce realistic human bodies. To this end, we employ the regularizer:

$$\mathcal{L}_\theta = \|\theta - \hat{\theta}\|^2. \tag{S.6}$$

Even after incorporating $\mathcal{L}_\theta$ into our optimization, we still observe some unrealistic results. In the left part of Fig. S.5, we illustrate such an example, where the body "tilts" unrealistically toward the object. To account for this, we introduce an additional regularization loss term:

$$\mathcal{L}_{reg} = cos^{-1}(\overrightarrow{\mathcal{FP}}\hat{\overrightarrow{\mathcal{FP}}}), \tag{S.7}$$

where $\mathcal{F}$ is the center of the feet, defined as the midpoint between the right and left ankle joints, and $\mathcal{P}$ the pelvis joint. By minimizing Eq. (S.7), we prevent deviations from the initial posture of our CReach, eliminating "tilting" effects.

To align the hand of the whole body (produced by CReach) with the one of the "guiding" hand grasp (produced by CGrasp) we use:

$$\mathcal{L}_{hm} = \|\mathcal{V}_{hm} - \hat{\mathcal{V}}_b^{hm}\|^2 + \lambda_{wrist}\|\mathcal{V}_{wrist} - \hat{\mathcal{V}}_b^{wrist}\|^2, \tag{S.8}$$

where $\mathcal{V}_{hm}$ are the hand vertices and $\mathcal{V}_{wrist}$ are the wrist vertices of the output hand mesh of CGrasp, and $\hat{\mathcal{V}}_b^{hm}$ and $\hat{\mathcal{V}}_b^{wrist}$ are the corresponding hand and wrist vertices of the optimized body's hand.

We use the Adam optimizer with a learning rate of 0.01. Our optimization has two stages. First, we optimize over all above losses for 800 iterations. Then, we exclude the penetration loss, and use the rest of the losses to optimize over the pose parameters of the shoulder, elbow, and wrist that correspond to the matching hand. The main goal of the second optimization step is to refine the hand grasp.

S.2. CWGrasp – Perceptual Study

To evaluate CWGrasp we conduct a perceptual study. Figure S.7 provides the task description that is shown to participants. Figure S.8 shows some samples used in our study.

Figure S.5. **Left:** Failure case when we do not use our regularizer term $\mathcal{L}_{reg}$ in our loss function. In some cases, when $\mathcal{L}_{reg}$ is not used, our model prefers to generate bodies that tilt unrealistically toward the object to prevent penetrations with the receptacle $\mathcal{M}$, while reaching the object with the hand. **Right:** The corresponding output of CWGrasp when we use our $\mathcal{L}_{reg}$ loss term.

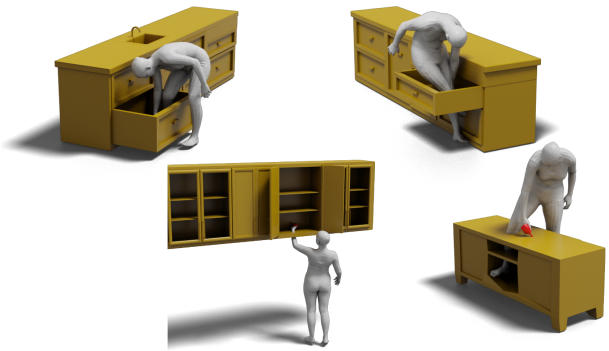

Figure S.6. **Failure cases** of our CWGrasp method for various receptacles. Our framework can fail for two reasons, namely due to penetrations between the body and the receptacle, or due to penetrations between the grasping hand and the object.

S.3. CWGrasp – Qualitative Evaluation

Here we provide additional qualitative comparisons between FLEX [54] and our CWGrasp framework; see Fig. S.9, Fig. S.10, and Fig. S.11.

For each example, we depict both a full-body view and a close-up view onto the hand and object. Often FLEX bodies miss contacting the object, or approach the object from unrealistic directions or have unrealistic orientations (e.g., they do not "look" at the object). Instead, CWGrasp produces significantly more realistic bodies and grasps, a finding supported also by our perceptual study.

Note that only CWGrasp generates both right- and left-hand interactions; for examples of the latter see Fig. S.12.

Figure S.13 shows results before and after optimization; we observe that optimization enhances realism significantly.

Failure cases: In Fig. S.6 we provide failure cases of our CWGrasp method. The reaching arm and hand look plausible, except for the bottom-left case where the arm penetrates the receptacle. The latter shows that occasionally sampling our ReachingField might fail, so subsequent optimization can be trapped in a local minimum, however, empirically, this doesn't happen often. In all other cases, the body or legs get trapped in a local minimum. This could be tackled with an involved modeling of the full scene, but this is out of our scope here, so we leave it for future work.

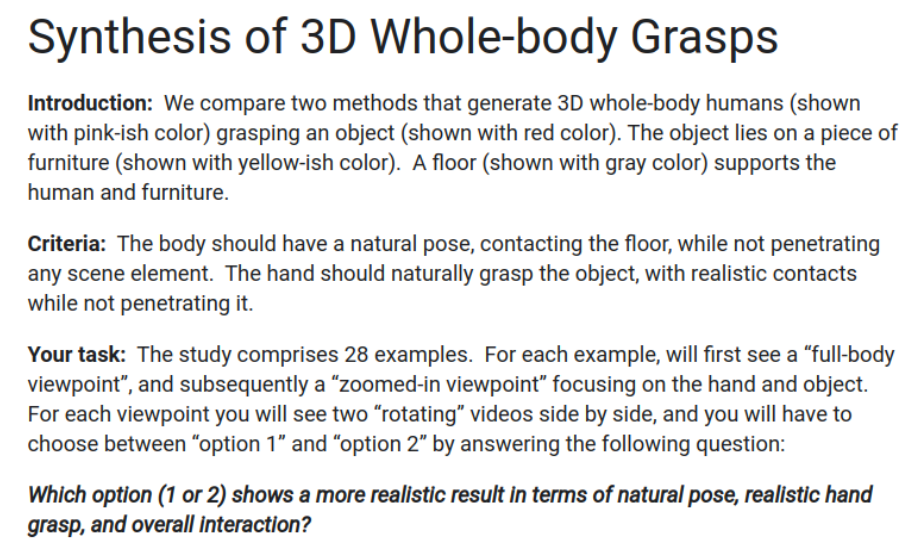

Figure S.7. **Perceptual study protocol.** We ask 35 participants to observe grasps generated by two methods for 28 object-and-receptacle configurations, and to specify which grasp is more realistic in terms of natural pose, realistic hand grasp, and overall interaction realism. This shows the task instructions shown to participants. See also the samples of Fig. S.8.

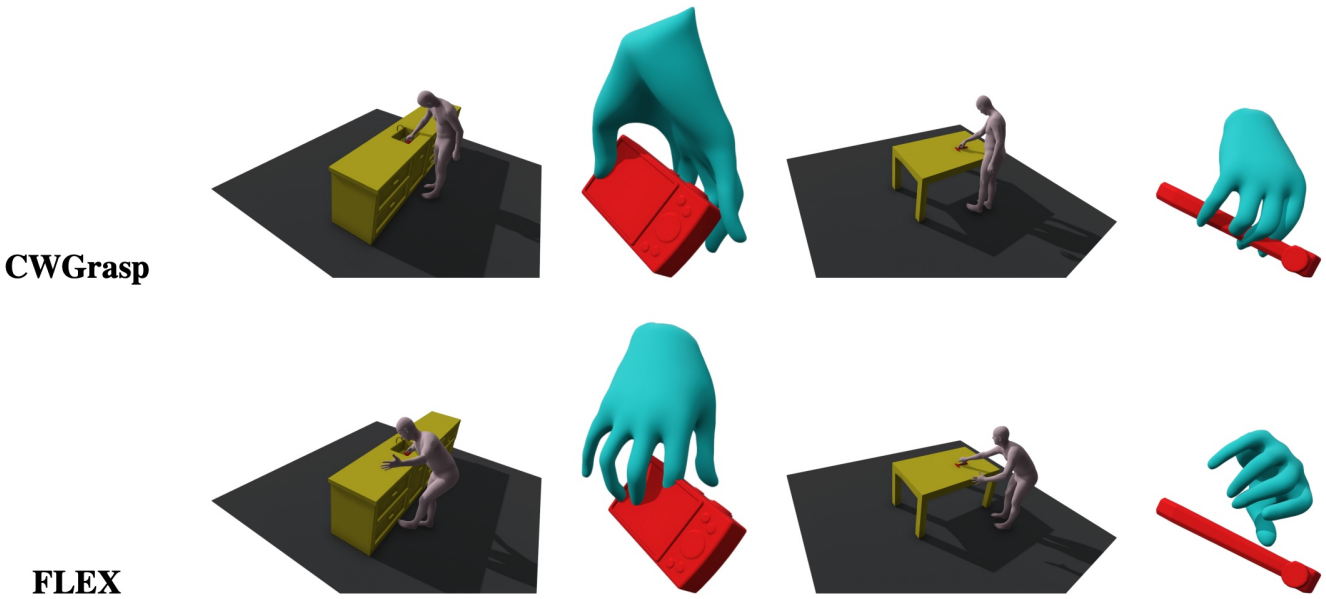

Figure S.8. **Perceptual study samples.** The first row shows grasps generated by our CWGrasp method, while the second row ones generated by FLEX [54]. For each sample, we show a full-body view and a close-up view onto the hand and object.

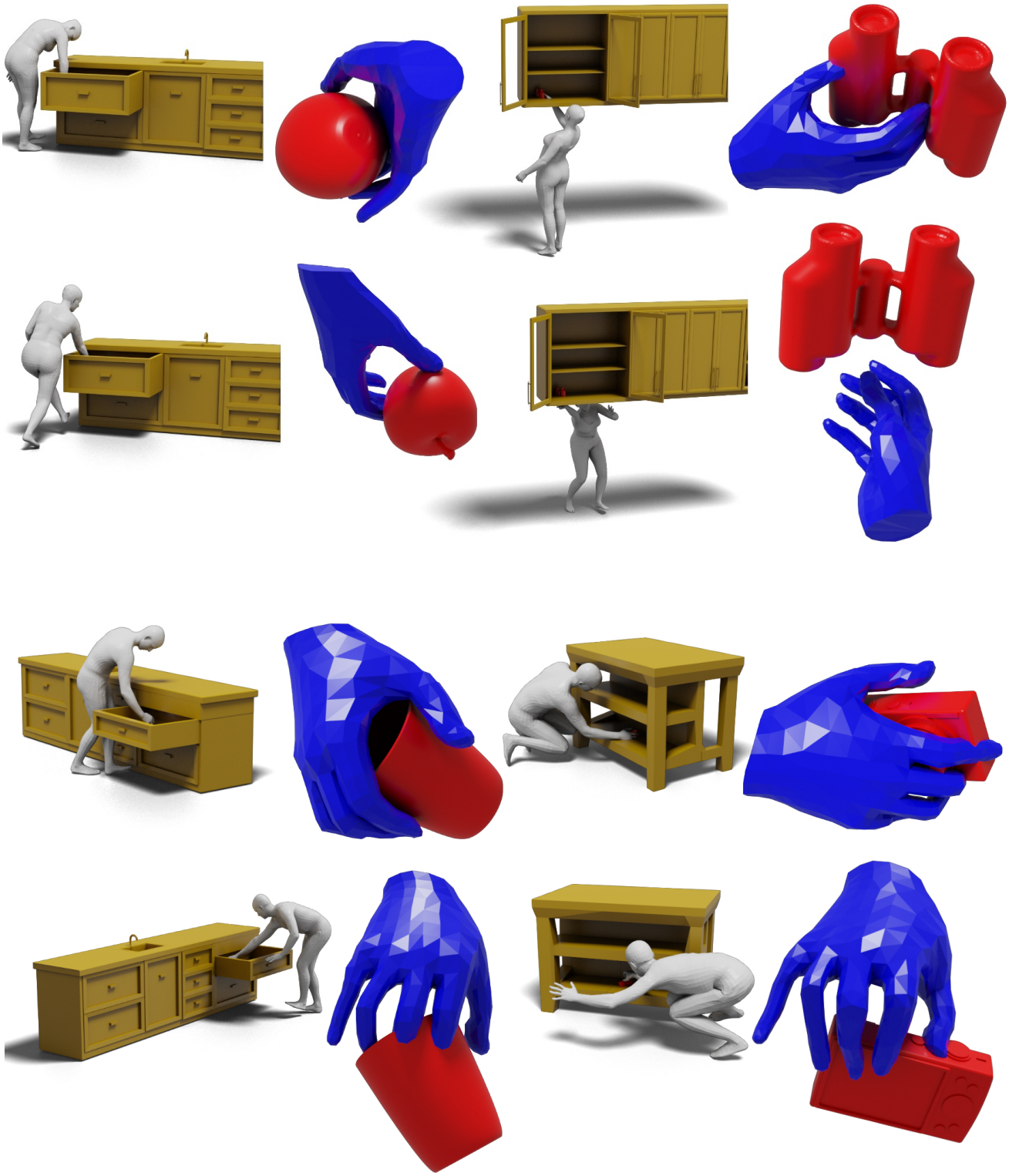

Figure S.9. Qualitative comparison of our CWGrasp method against FLEX [54]. The **first** and **third** rows show results generated by CWGrasp, while the **second** and **fourth** ones show results generated by FLEX, for the same object and receptacle configurations. For each grasp we depict both the full-body (with gray meshes) and "scene," as well as a close-up onto the hand and object (with blue and red meshes, respectively, from a side view). For FLEX, out of all its generated samples, we visualize the "best" one, i.e., the one with the smallest total loss. In contrast, our CWGrasp generates only one sample.

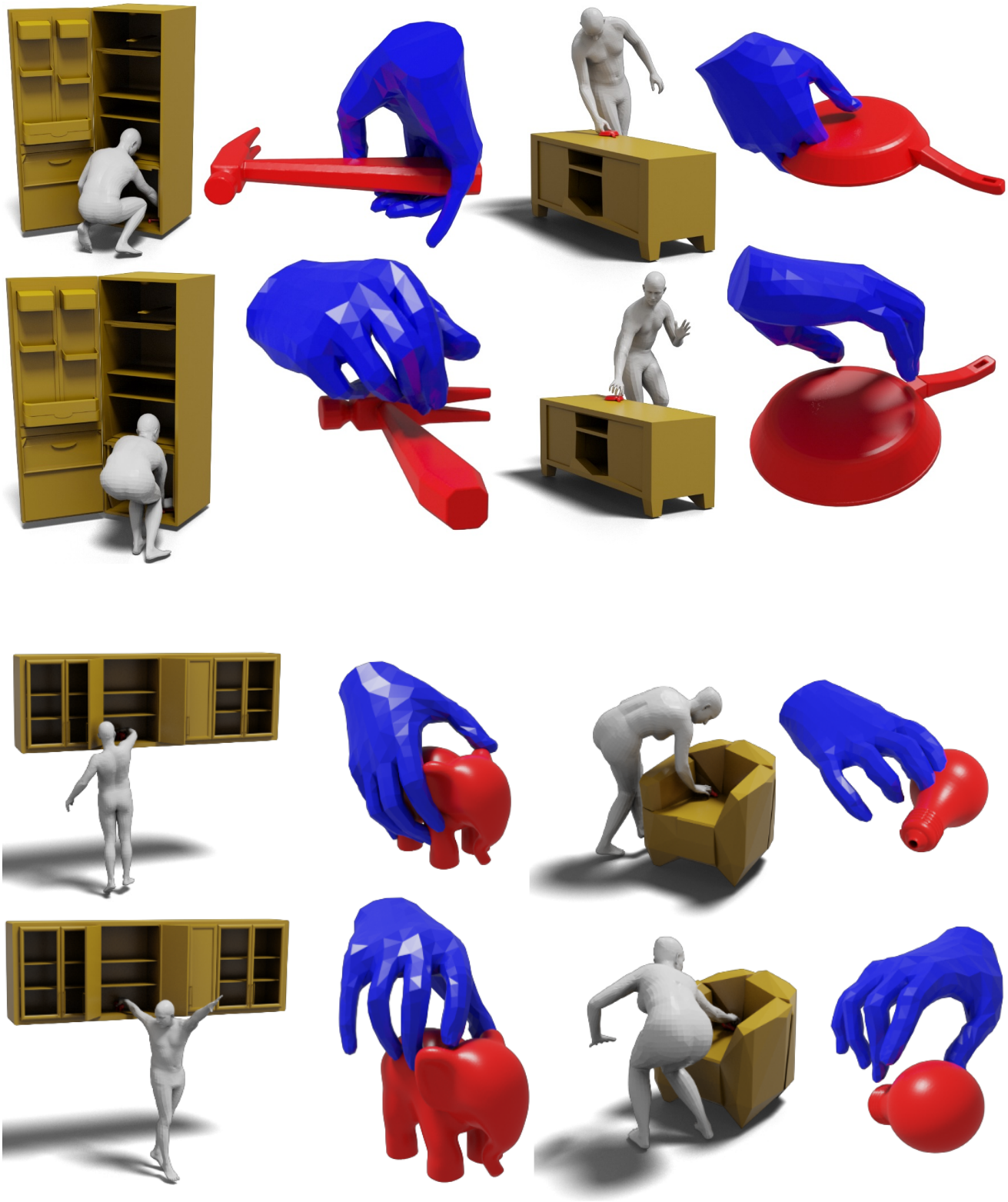

Figure S.10. Qualitative comparison of our CWGrasp method against FLEX [54]. The **first** and **third** rows show results generated by CWGrasp, while the **second** and **fourth** ones show results generated by FLEX, for the same object and receptacle configurations. For each grasp we depict both the full-body (with gray meshes) and "scene," as well as a close-up onto the hand and object (with blue and red meshes, respectively, from a side view). For FLEX, out of all its generated samples, we visualize the "best" one, i.e., the one with the smallest total loss. In contrast, our CWGrasp generates only one sample.

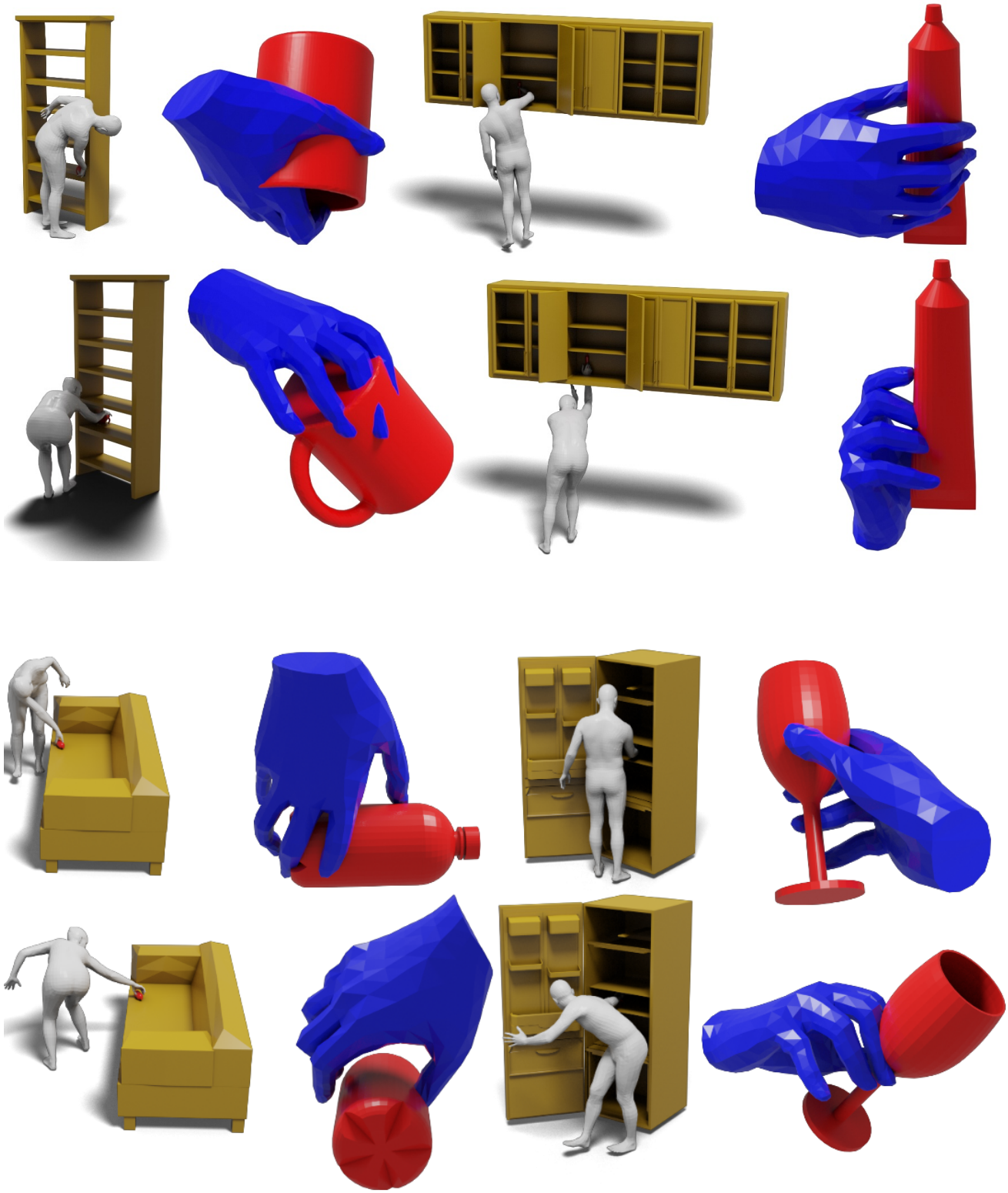

Figure S.11. Qualitative comparison of our CWGrasp method against FLEX [54]. The **first** and **third** rows show results generated by CWGrasp, while the **second** and **fourth** ones show results generated by FLEX, for the same object and receptacle configurations. For each grasp we depict both the full-body (with gray meshes) and "scene," as well as a close-up onto the hand and object (with blue and red meshes, respectively, from a side view). For FLEX, out of all its generated samples, we visualize the "best" one, i.e., the one with the smallest total loss. In contrast, our CWGrasp generates only one sample.

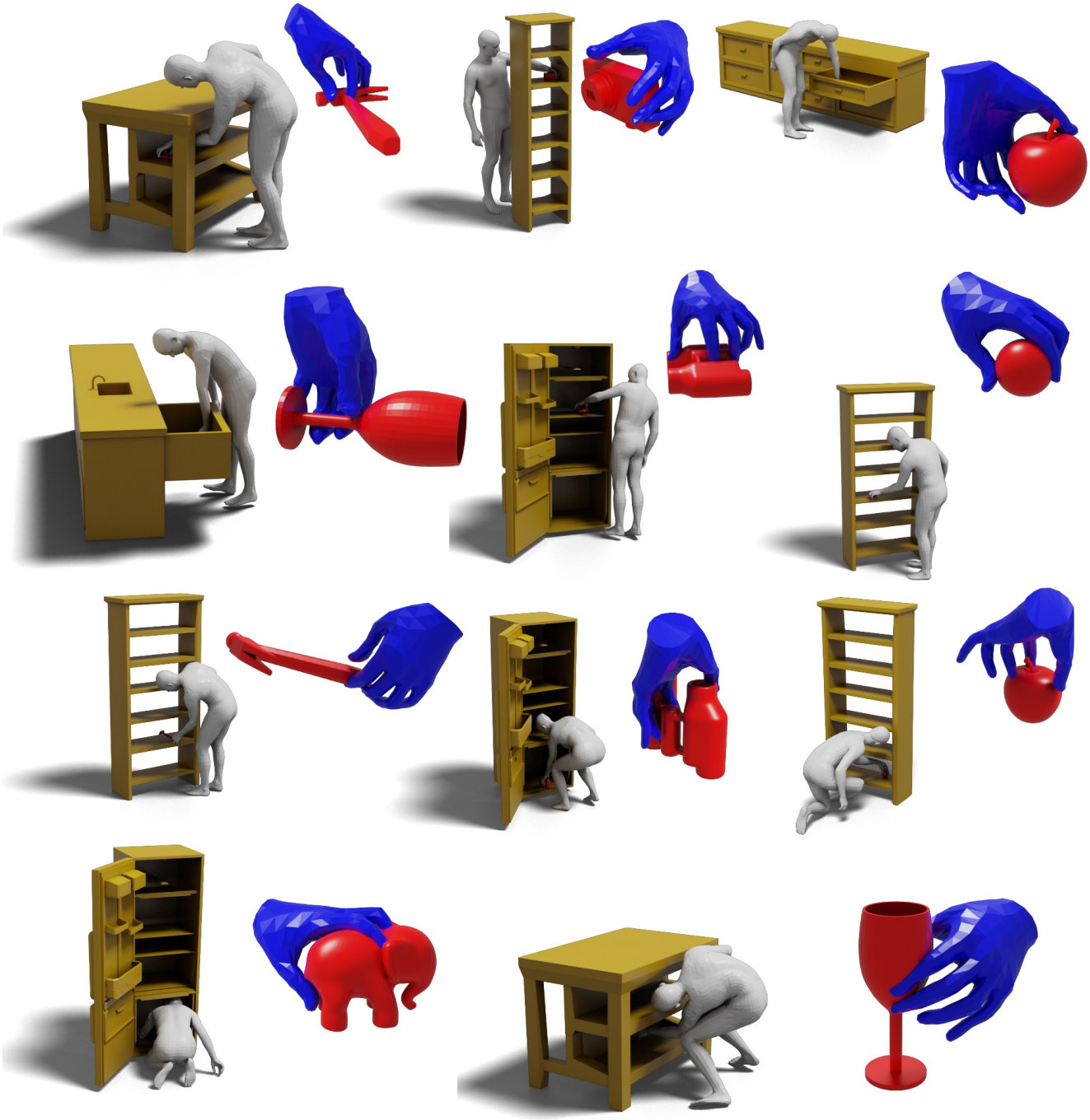

Figure S.12. Qualitative results of CWGrasp when using the left hand. Note that CWGrasp is unique in generating both right-hand and left-hand whole-body grasps, while performance is similar for both cases.

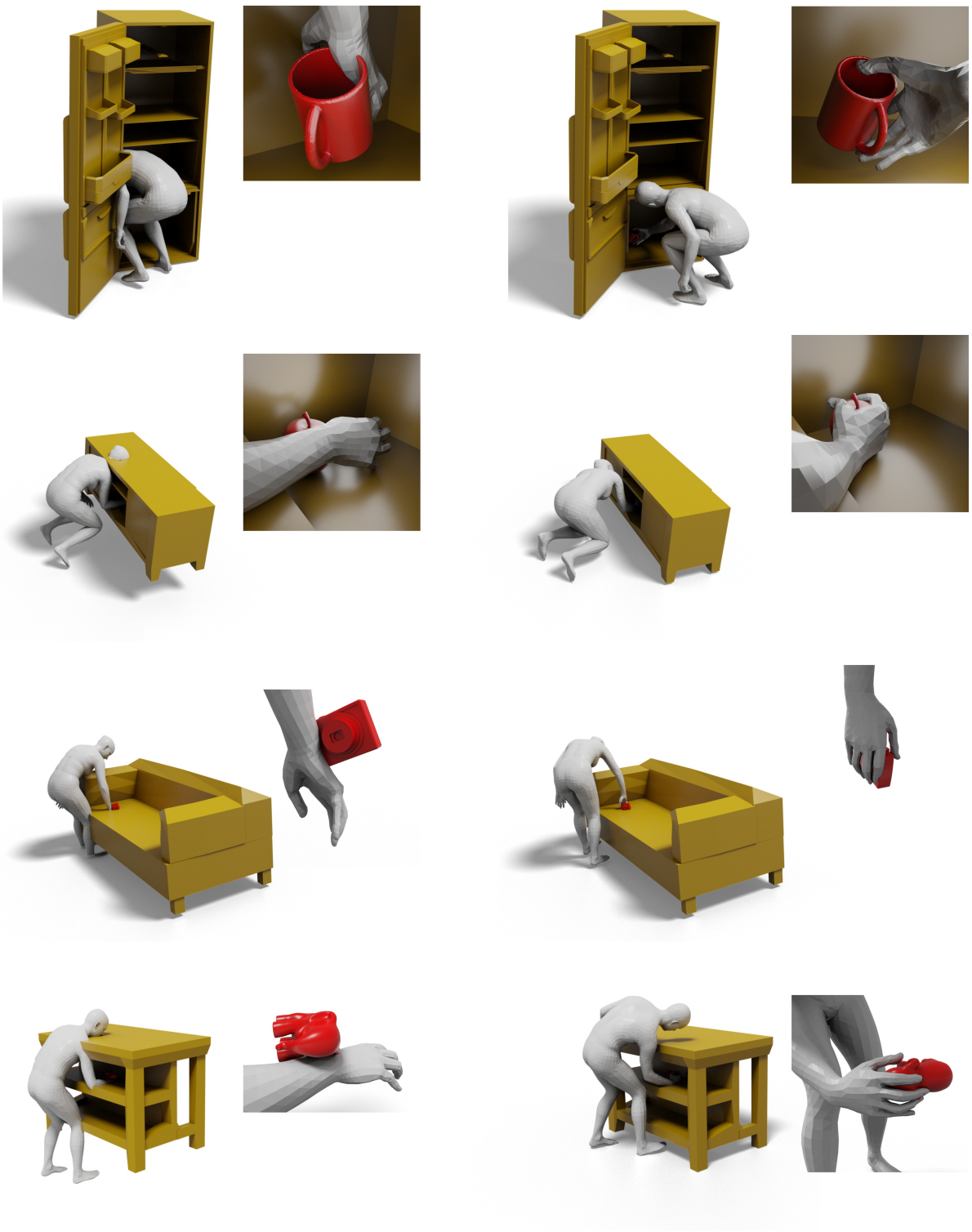

Figure S.13. Qualitative evaluation of CWGrasp optimization performance. We show the generated interacting bodies before (left) and after (right) optimization with CWGrasp. For each example we show both a full-body view and a close-up view on the hand and object.