

## 587 A Data Process 615

### 588 A.1 Point Cloud Denoising

589 Hardware limitations of RGBD cameras can introduce noise  
 590 in the point clouds they generate, which may affect the ac-  
 591 curacy of pose annotation. We refine our point cloud data by  
 592 applying a statistical outlier removal filter [Zhou *et al.*, 2018].  
 593 This process involves analyzing each point’s average distance  
 594 to its 20 nearest neighbors and excluding those points whose  
 595 distance deviates by more than two standard deviations from  
 596 the mean, effectively reducing noise.

### 597 A.2 Object Pose Labeling

598 The 6D poses of objects are annotated mainly using the It-  
 599 erative Closet Point method (ICP) [Besl and McKay, 1992]  
 600 with human adjustment. Initially We manually determine the  
 601 object pose in the first frame using the refined point cloud,  
 602 setting the foundation for subsequent automated ICP adjust-  
 603 ments. The pose for each subsequent frame is inferred from  
 604 the preceding one. Finally, the resulting sequence is inspected  
 605 and, if necessary, fine-tuned by a human annotator. In prac-  
 606 tice, most sequences require only a single annotation pass.

### 607 A.3 Visualization of Dataset

608 **Annotation** Here we present a sample of the annotated results  
 609 depicting the object motion and dexterous hand motion, as  
 shown in Figure 7.

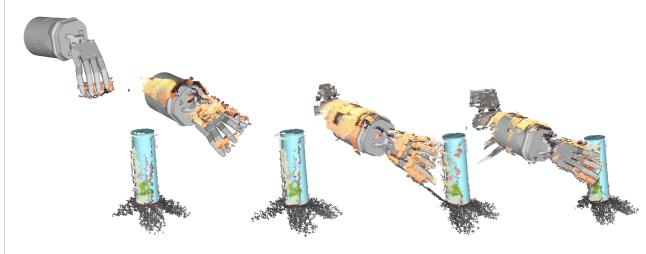


Figure 7: The visualization for aligned point cloud and hand’s mesh, object’s mesh.

610 **Motion Sequence** We present the the motion sequence of  
 611 dexterous hand mesh in our RealDex dataset. We sampled  
 612 8 frames from a grasping motion and display the mesh of  
 613 robotic hand with arm, as shown in Figure 8.

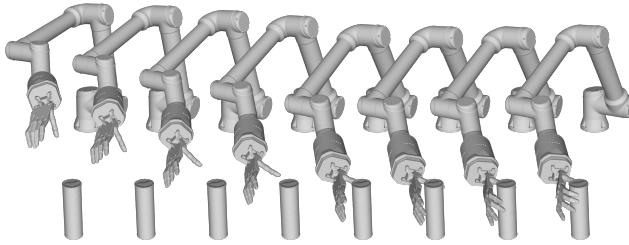


Figure 8: The visualization for grasping motion sequence in Re-  
 alDex.

## 614 B Method 615

### 615 B.1 Training 616

We train our framework in two stages, the training for grasp  
 617 pose generation and the training for motion synthesis. Since  
 618 our dataset includes precise annotations for object and hand  
 619 poses along with complete dexterous hand motion, enables  
 620 both stages of our training to benefit from ground truth data  
 621 supervision.

**Pose Generation** During pose generation training, we first  
 622 create the robotic hand’s mesh from the hand pose  $\phi$  using  
 623 forward kinematics and then generate the hand’s point cloud  
 624  $\mathbf{P}^h$ . The hand feature  $\mathcal{F}^h$  and condition feature  $\mathcal{F}^o$  is com-  
 625 pressed into the latent space by cVAE encoder. Hand poses  
 626 are reconstructed by the decoder using the concatenation of  
 627 conditional feature and the latent code, sampled from the  
 628 learned distribution. From the decoder’s output, we can then  
 629 compute a binary contact map,  $\mathcal{C}$  on object points that indi-  
 630 cates whether the points are within the hand’s contact region.  
 631 The loss to supervise the generated poses is the weighted sum  
 632 of four losses:

$$\begin{aligned}\mathcal{L}_{\text{KL}} &= \frac{1}{2}(-\log \sigma^2 - 1 + \sigma^2 + \mu^2) \\ \mathcal{L}_{\text{recon}} &= \|\phi - \phi^{\text{gt}}\|_2, \\ \mathcal{L}_{\text{cmap}} &= \text{BCE}(\mathcal{C} - \mathcal{C}^{\text{gt}}), \\ \mathcal{L}_{\text{CD}} &= \sum_{\mathbf{a} \in \mathbf{P}^h} \min_{\mathbf{b} \in \mathbf{P}^{h,\text{gt}}} \|\mathbf{a} - \mathbf{b}\|^2 + \sum_{\mathbf{b} \in \mathbf{P}^{h,\text{gt}}} \min_{\mathbf{a} \in \mathbf{P}^h} \|\mathbf{b} - \mathbf{a}\|^2.\end{aligned}\quad (4)$$

In Equation 4,  $\mathcal{L}_{\text{KL}}$  denotes the Kullback-Leibler divergence  
 636 to measure the similarity between prior  $\mathcal{N}(\mu, \sigma^2)$  and stan-  
 637 dard Gaussian distribution  $\mathcal{N}(0, 1)$ ;  $\mathcal{L}_{\text{recon}}$  is the MSE loss of  
 638 reconstructed hand pose and ground truth hand pose;  $\mathcal{L}_{\text{cmap}}$   
 639 is a binary cross entropy (BCE) to measure the difference be-  
 640 tween the contact map from reconstructed hand pose and the  
 641 ground truth; and  $\mathcal{L}_{\text{CD}}$  is the Chamfer distance between points  
 642 sampled from reconstruction hand mesh and the points on GT  
 643 hand mesh.

**Motion Synthesis** In the training of MotionNet, we first gen-  
 644 erate the hand points  $\mathbf{P}^h$ . Then we add noise to  $\phi$  and  $\mathbf{P}^h$   
 645 in the network input to enhance the generalization ability of  
 646 network. The loss for MotionNet is the difference from pre-  
 647 dicted parameters to its GT value.

$$\mathcal{L}_M = \omega_\phi \|\phi - \phi^{\text{gt}}\|_1 + \omega_h \|\mathbf{P}^h - \mathbf{P}^{h,\text{gt}}\|_2 + \omega_d \|\mathbf{d}^h - \mathbf{d}^{h,\text{gt}}\|_2 \quad (5)$$

### 650 B.2 MLLM Selection

For each object, we sample 100 poses and generate 100 im-  
 651 ages through rendering. These images are collectively pro-  
 652 cessed by Gemini, yielding a set of scores along with de-  
 653 tails explanations for each pose. Subsequently, we extract  
 654 the top ten poses from the dataset, which are determined by  
 655 the scores they received. These selected poses serve as the  
 656 primary targets for our subsequent motion synthesis phase.

Please rate the quality of the grasp shown in this picture, the score should be out of 5. Before assigning a score, consider the following questions:

- Does it grasp an object?
- Which part of the object does it grasp?
- Is there any special functionality associated with this part?
- Which fingers are being used to grasp the object?
- Is the grasp stable?
- Does the hand pose look natural and in line with typical human habits?

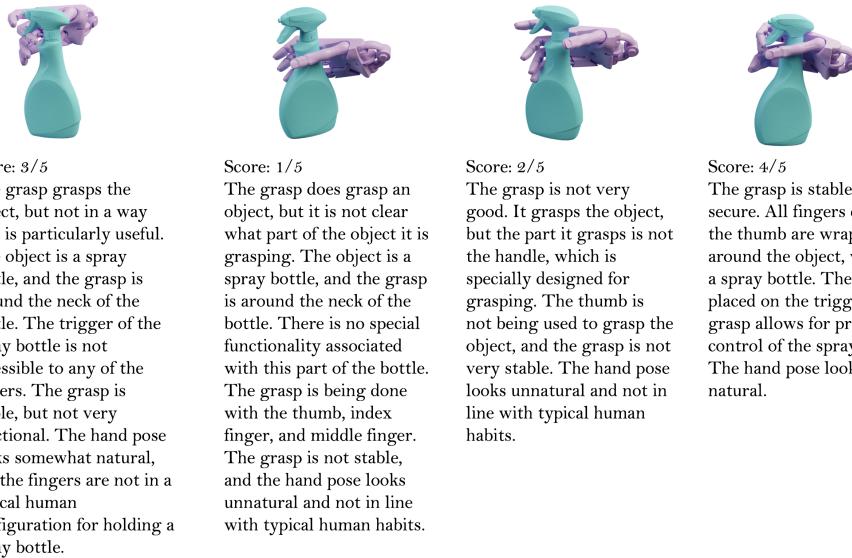


Figure 9: The text in the first column provides the complete prompt input to Gemini. Adjacent to this, in the subsequent four columns on the right, we present the input images alongside the corresponding scores and explanations as given by the MLLM selection module, offering a transparent view of the decision-making process.

## B.3 Inference

At the inference stage, our pose generation module receives unseen object point clouds, which serve as the input conditions. Utilizing these conditions, cVAE decoder generates candidate grasping by randomly sampling the latent code from standard Gaussian distribution. Candidate poses are refined by test-time optimization and then get scores from LLM selection module, special requirements or conditions can be added to let the LLM select the most suitable pose as goal. Finally the MotionNet utilizes the selected poses as targets and initiates the motion synthesis process from the mean pose, indicating that all joint angles of the dexterous hand are set to zero. The output for the current time frame is then employed to determine the input data for the subsequent time frame. The termination of this process is defined by either fixed time steps or a threshold based on the distance between the current grasp and the target grasp.

## C More Results and Discussions

### C.1 Pose generation

**Figure 11** displays selected results from our grasping pose generation module, showcasing various automatically computed hand configurations for different object shapes.

## C.2 Motion synthesis

Given a initial pose and a target pose, our pose-guided hand motion synthesis module is capable of generating a sequence of hand motion, as shown in [Figure 10](#), the initial pose we give is the mean pose of dexterous hand, which means that all the joint angles equal 0 in this pose. The translation of the hand is calculated from the average location across our dataset. Each one in the generated sequence represents a progressive step towards achieving the final target configuration.

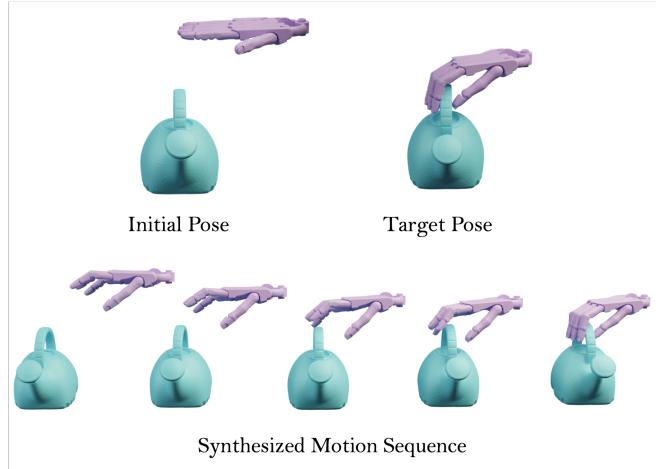


Figure 10: Motion synthesis result from our framework. The first row illustrates the initial and target hand poses, serving as inputs for the motion synthesis module. Subsequently, a sequence of hand motions is generated, using the target pose as a reference to guide the synthesis process.

### C.3 MLLM selection

In Figure 9, we show the output from our MLLM selection module, each grasp is represented by a rendered image of the hand and object mesh. These images are input into the MLLM selection module, which assigns a score to each grasp and give detailed explanation.

## D Limitation

Our algorithm still has much room for improvement. For instance, in the result of pose generation, there is intersec-

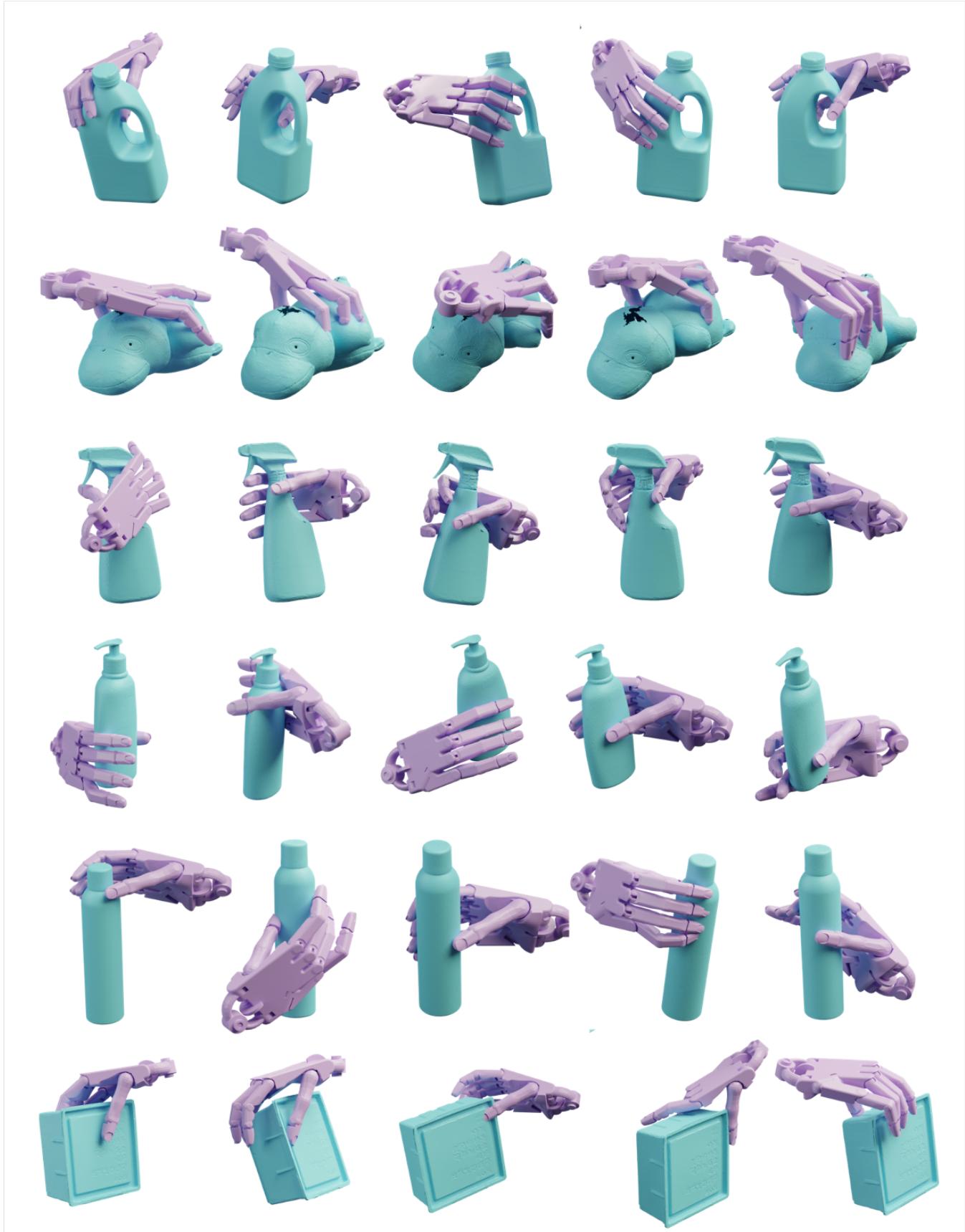


Figure 11: Visualization of the generated grasps from our grasp pose generation module. Given an object point cloud derived from RGB-D data, this module samples potential hand poses and employs MLLM to select the most plausible ones.

698 tion between object and hand that need to be removed by  
699 optimization in test time. It could be improved by utilizing  
700 penalty loss for collision when training. In addition, when  
701 generating motion, it is guided solely by the target pose, with-  
702 out taking into account the actual conditions of the objects  
703 and the environment.