

587 A Data Process

588 A.1 Point Cloud Denoising

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

597 A.2 Object Pose Labeling

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

607 A.3 Visualization of Dataset

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

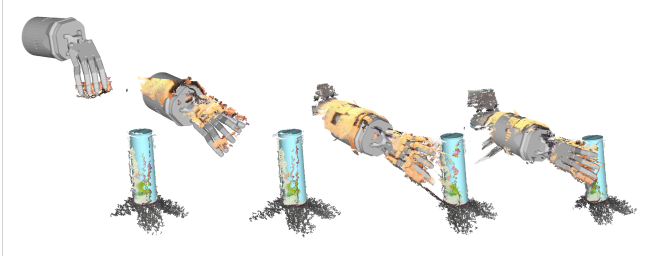


Figure 7: The visualization for aligned point cloud and hand’s mesh, object’s mesh.

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

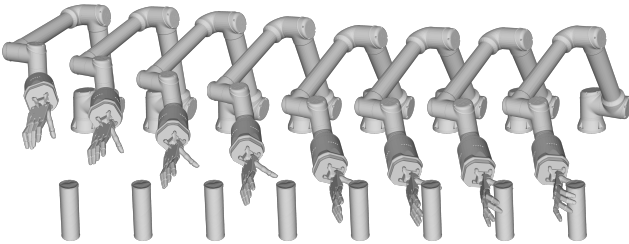


Figure 8: The visualization for grasping motion sequence in RealDex.

B Method

B.1 Training

615 We train our framework in two stages, the training for grasp pose generation and the training for motion synthesis. Since our dataset includes precise annotations for object and hand poses along with complete dexterous hand motion, enables both stages of our training to benefit from ground truth data supervision. 622 623

624 **Pose Generation** During pose generation training, we first create the robotic hand’s mesh from the hand pose ϕ using forward kinematics and then generate the hand’s point cloud \mathbf{P}^h . The hand feature \mathcal{F}^h and condition feature \mathcal{F}^o is compressed into the latent space by cVAE encoder. Hand poses are reconstructed by the decoder using the concatenation of conditional feature and the latent code, sampled from the learned distribution. From the decoder’s output, we can then compute a binary contact map, \mathcal{C} on object points that indicates whether the points are within the hand’s contact region. The loss to supervise the generated poses is the weighted sum of four losses: 634 635

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

In Equation 4, \mathcal{L}_{KL} denotes the Kullback-Leibler divergence to measure the similarity between prior $\mathcal{N}(\mu, \sigma^2)$ and standard Gaussian distribution $\mathcal{N}(0, 1)$; $\mathcal{L}_{\text{recon}}$ is the MSE loss of reconstructed hand pose and ground truth hand pose; $\mathcal{L}_{\text{cmap}}$ is a binary cross entropy (BCE) to measure the difference between the contact map from reconstructed hand pose and the ground truth; and \mathcal{L}_{CD} is the Chamfer distance between points sampled from reconstruction hand mesh and the points on GT hand mesh. 644

645 **Motion Synthesis** In the training of MotionNet, we first generate the hand points \mathbf{P}^h . Then we add noise to ϕ and \mathbf{P}^h in the network input to enhance the generalization ability of network. The loss for MotionNet is the difference from predicted parameters to its GT value. 648 649

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

B.2 MLLM Selection

650 For each object, we sample 100 poses and generate 100 images through rendering. These images are collectively processed by Gemini, yielding a set of scores along with detailed explanations for each pose. Subsequently, we extract the top ten poses from the dataset, which are determined by the scores they received. These selected poses serve as the primary targets for our subsequent motion synthesis phase. 657

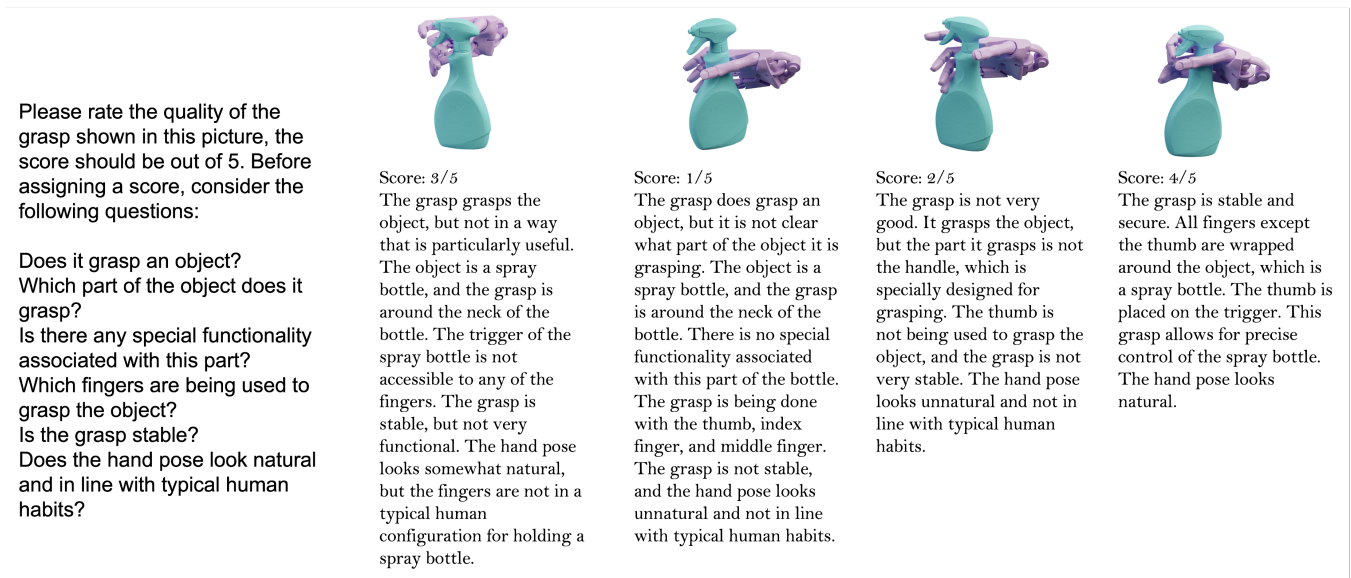


Figure 9: The text in the first column provides the complete prompt input to Gemini. Adjacent to this, in the subsequent four columns on the right, we present the input images alongside the corresponding scores and explanations as given by the MLLM selection module, offering a transparent view of the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

658 B.3 Inference

659 At the inference stage, our pose generation module receives
 660 unseen object point clouds, which serve as the input condi-
 661 tions. Utilizing these conditions, cVAE decoder generates
 662 candidate grasping by randomly sampling the latent code
 663 from standard Gaussian distribution. Candidate poses are re-
 664 fined by test-time optimization and then get scores from LLM
 665 selection module, special requirements or conditions can be
 666 added to let the LLM select the most suitable pose as goal. Fi-
 667 nally the MotionNet utilizes the selected poses as targets and
 668 initiates the motion synthesis process from the mean pose, in-
 669 dicating that all joint angles of the dexterous hand are set to
 670 zero. The output for the current time frame is then employed
 671 to determine the input data for the subsequent time frame.
 672 The termination of this process is defined by either fixed time
 673 steps or a threshold based on the distance between the current
 674 grasp and the target grasp.

675 C More Results and Discussions

676 C.1 Pose generation

677 **Figure 11** displays selected results from our grasping pose
 678 generation module, showcasing various automatically com-
 679 puted hand configurations for different object shapes.

680 C.2 Motion synthesis

681 Given an initial pose and a target pose, our pose-guided hand
 682 motion synthesis module is capable of generating a sequence
 683 of hand motion, as shown in **Figure 10**, the initial pose we
 684 give is the mean pose of dexterous hand, which means that
 685 all the joint angles equal 0 in this pose. The translation of
 686 the hand is calculated from the average location across our
 687 dataset. Each one in the generated sequence represents a pro-
 688 gressive step towards achieving the final target configuration.

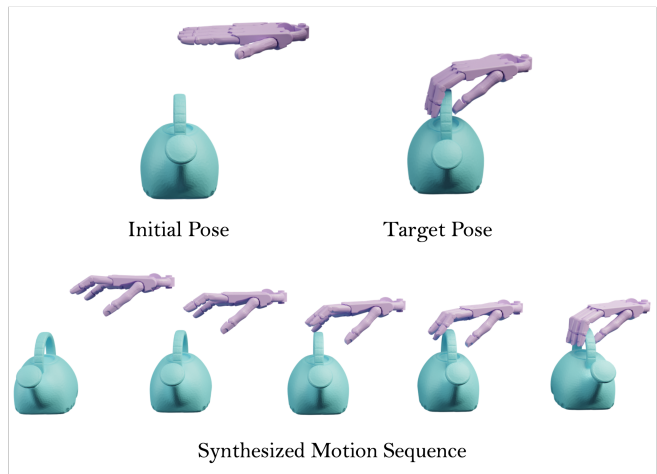


Figure 10: Motion synthesis result from our framework. The first row illustrates the initial and target hand poses, serving as inputs for the motion synthesis module. Subsequently, a sequence of hand motions is generated, using the target pose as a reference to guide the synthesis process.

689 C.3 MLLM selection

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

695 D Limitation

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

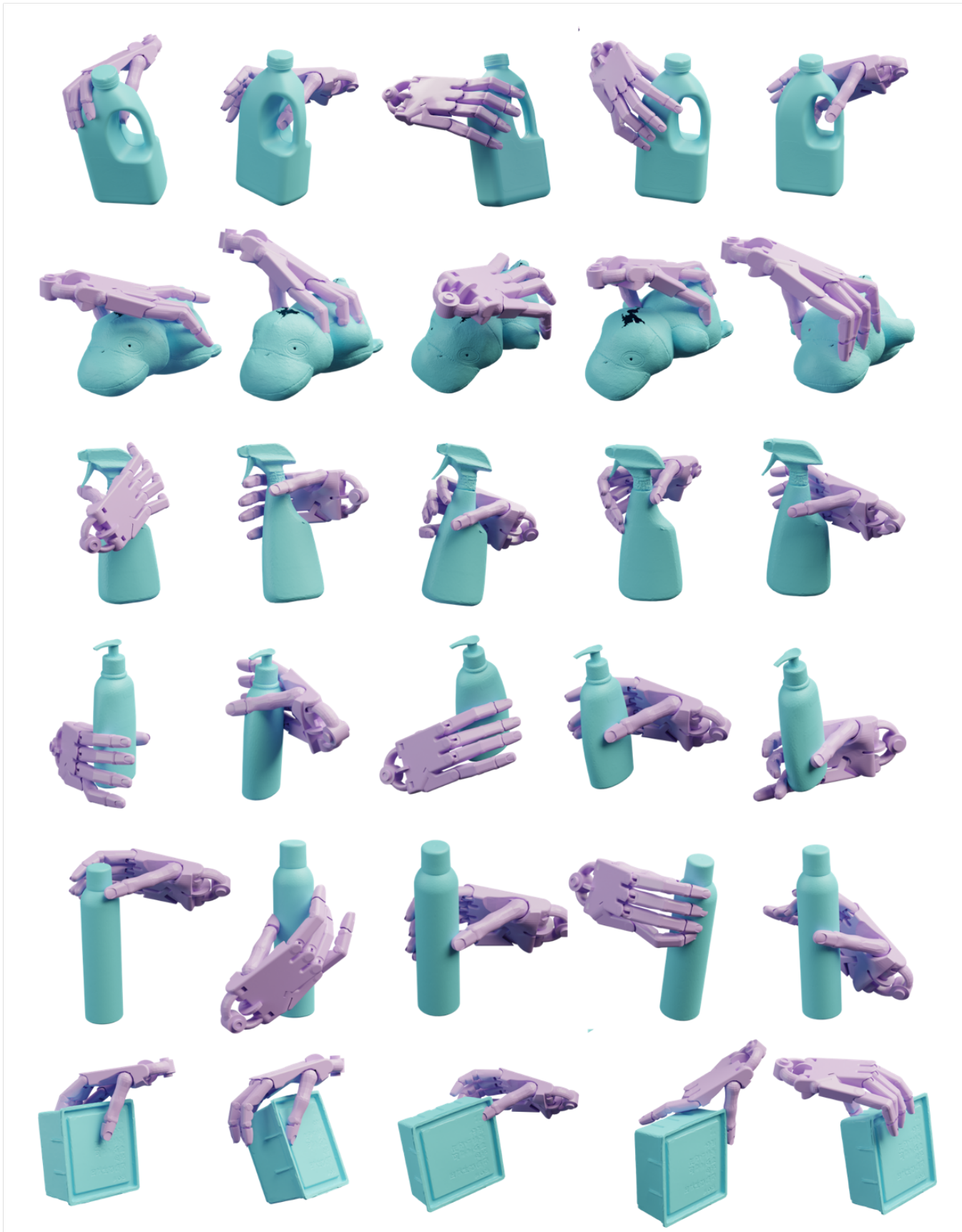


Figure 11: Visualization of the generated grasps from our grasp pose generation module. Given an object point cloud derived from RGB-D data, this module samples potential hand poses and employs MLLM to select the most plausible ones.

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.