CAMS: CAnonicalized Manipulation Spaces for Category-Level Functional Hand-Object Manipulation Synthesis

Juntian Zheng[*,1,3] Qingyuan Zheng[*,3] Lixing Fang[*,1,3] Yun Liu[1] Li Yi[†,1,2,3]
[1]IIIS, Tsinghua University [2]Shanghai Artificial Intelligence Laboratory
[3]Shanghai Qi Zhi Institute

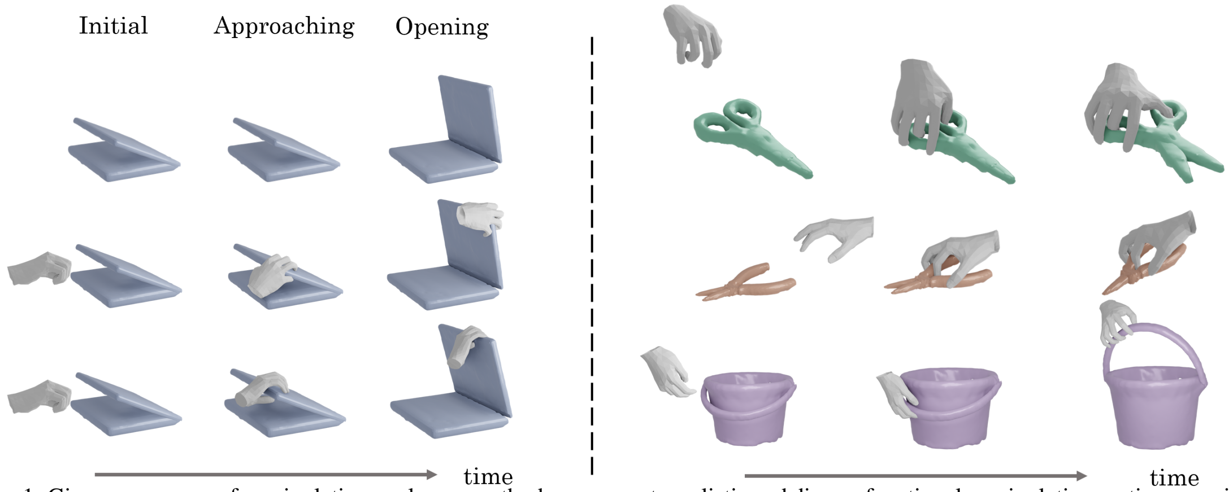


Figure 1. Given a sequence of manipulation goals, our method can generate realistic and diverse functional manipulation motions consistent with the goal sequence. The motions are expressed in snapshots at several keyframes. On the left side, we show a goal sequence of opening a laptop and two different manipulation patterns that can be generated by our method. On the right side, we show our generation results of three manipulation tasks corresponding to different object categories.

Abstract

In this work, we focus on a novel task of category-level functional hand-object manipulation synthesis covering both rigid and articulated object categories. Given an object geometry, an initial human hand pose as well as a sparse control sequence of object poses, our goal is to generate a physically reasonable hand-object manipulation sequence that performs like human beings. To address such a challenge, we first design CAnonicalized Manipulation Spaces (CAMS), a two-level space hierarchy that canonicalizes the hand poses in an object-centric and contact-centric view. Benefiting from the representation capability of CAMS, we then present a two-stage framework for synthesizing human-like manipulation animations. Our framework achieves state-of-the-art performance for both rigid and articulated categories with impressive visual effects. Codes and video results can be found at our project homepage: `https://cams-hoi.github.io/`.

*Equal contribution with the order determined by rolling dice.
†Corresponding author.

1. Introduction

Human conducts hand-object manipulation (HOM) for certain functional purposes commonly in daily life, *e.g.* opening a laptop and using scissors to cut. Understanding how such manipulation happens and being able to synthesize realistic hand-object manipulation has naturally become a key problem in computer vision. A generative model that can synthesize human-like functional hand-object manipulation plays an essential role in various applications, including video games, virtual reality, dexterous robotic manipulation, and human-robot interaction.

This problem has only been studied with a very limited scope previously. Most existing works focus on the synthesis of a static grasp either with [3] or without [17] a functional goal. Recently, there have been works started focusing on dynamic manipulation synthesis [7, 41]. However, these works restrict their scope to rigid objects and do not consider the fact that functional manipulation might change the object geometry as well, such as in opening a laptop by hand. Moreover, these works usually require a strong input, including hand and object trajectories or a grasp reference, limiting their application scenarios.

To expand the scope of HOM synthesis, we propose a new task of category-level functional hand-object manipulation synthesis. Given a 3D shape from a known category as well as a sequence of functional goals, our task is to synthesize human-like and physically realistic hand-object manipulation to sequentially realize the goals as shown in Figure 1. Besides rigid objects, we also consider articulated objects, which support richer manipulations than a simple move. We represent a functional goal as a 6D pose for each rigid part of the object. We emphasize category-level for generalization to unseen geometry and for more human-like manipulations revealing the underlying semantics.

In this work, we choose to tackle the above task with a learning approach. We can learn from human demonstrations for HOM synthesis thanks to the recent effort in capturing category-level human-object manipulation dataset [25]. The key challenges lie in three aspects. First, a synthesizer needs to generalize to a diverse set of geometry with complex kinematic structures. Second, humans can interact with an object in diverse ways. Faithfully capturing such distribution and synthesizing in a similar manner is difficult. Third, physically realistic synthesis requires understanding the complex dynamics between the hand and the object. Such understanding makes sure that the synthesized hand motion indeed drives the object state change without violating basic physical rules.

To address the above challenges, we choose to generate object motion through motion planning and learn a neural synthesizer to generate dynamic hand motion accordingly. Our key idea is to canonicalize the hand pose in an object-centric and contact-centric view so that the neural synthesizer only needs to capture a compact distribution. This idea comes from the following key observations. During functional hand-object manipulation, human hands usually possess a strong preference for the contact regions, and such preference is highly correlated to the object geometry, *e.g*. hand grasping the display edge while opening a laptop. From the contact point's perspective, the finger pose also lies in a low-dimensional space. Representing hand poses from an object-centric view as a set of contact points and from a contact-centric view as a set of local finger embeddings could greatly reduce the learning complexity.

Specifically, given an input object plus several functional goals, we first interpolate per-part object poses between every adjacent functional goal, resulting in an object motion trajectory. Then we take a two-stage method to synthesize the corresponding hand motion. In the first stage, we introduce CAnonicalized Manipulation Spaces (CAMS) to plan the hand motion. CAMS is defined as a two-level space hierarchy. At the root level, all corresponding parts from the category of interest are scale-normalized and consistently oriented so that the distribution of possible contact points becomes concentrated. At the leaf level, each contact point would define a local frame. This local frame would simplify the distribution of the corresponding finger pose. With CAMS, we could represent a hand pose as an object-centric and contact-centric CAMS embedding. At the core of our method is a conditional variation auto-encoder, which learns to predict a CAMS embedding sequence given an object motion trajectory. In the second stage, we introduce a contact- and penetration-aware motion synthesizer to further synthesize an object motion-compatible hand motion given the CAMS embedding sequence.

To summarize, our main contributions include: i) A new task of functional category-level hand-object manipulation synthesis. ii) CAMS, a hierarchy of spaces canonicalizing category-level HOM enabling manipulation synthesis for unseen objects. iii) A two-stage motion synthesis method to synthesize human-like and physically realistic HOM. iv) State-of-the-art HOM synthesis results for both articulated and rigid object categories.

2. Related Work

2.1. Human Motion Synthesis

Human motion synthesis, including motion prediction, interpolation, and completion, has attracted many interests these years [1, 2, 4, 8, 9, 12, 14–16, 20, 21, 26, 30, 37, 38, 42–44]. Conditional variational autoencoder [45] (CVAE) was widely used [4, 26, 30, 43] for its generalizability across various scenes and human motions. These works have achieved great success in human motion synthesis, while their potential in HOM synthesis was overlooked. In particular, [8, 37, 42] focused on modeling human-scene interaction when generating human motion. Such scene-aware or context-aware methods might also suit HOM synthesis.

2.2. Physics-Based Object Manipulation Synthesis

Generating high-quality human grasps remains challenging due to the complex geometry and complicated skeletal constraints. Physics-based methods [7, 23, 28, 33, 39, 40] were favored for in-hand manipulation synthesis since the generated motion was physically plausible. IBS [33] presented a novel representation of hand-object interaction and leveraged reinforcement learning (RL) methods with execution success, and geometric measure [11] rewards to generate successful grasping motion. D-Grasp [7] developed its grasping policy based on the physical attributes of hand and object, including angles and velocities. Yang *et al*. [39] concentrated on using chopsticks in diverse gripping styles, and it solved this rather difficult task by first optimizing physically valid gripping poses with predefined gripping styles and then utilizing the carefully designed hand-controlled policies to synthesize manipulation.

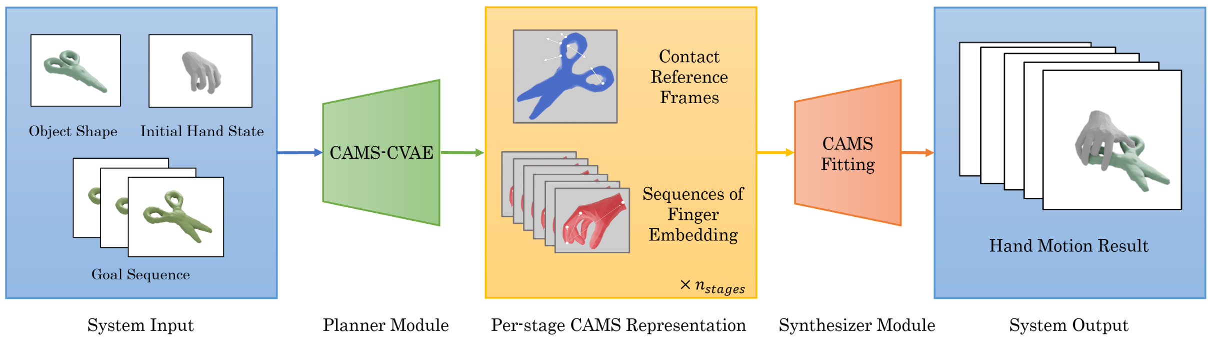


Figure 2. **System Overview.** Our framework mainly consists of a CVAE-based planner module and an optimization-based synthesizer module. Given the generation condition as the input, the planner first generates a per-stage CAMS representation containing contact reference frames and sequences of finger embedding. Then the synthesizer optimizes the whole manipulation animation based on the CAMS embedding.

2.3. Data-Driven Object Manipulation Synthesis

Besides physics-based methods, a line of data-driven approaches [5, 6, 10, 13, 17, 18, 24, 34, 35, 41] could generate manipulation in a more natural and human-like manner and generalize to novel object instances. Most of the previous data-driven works focused on reconstruction and synthesis for the static grasp [5, 6, 10, 13, 17, 18, 24, 35]. Grasping Field [18] and CPF [6] reconstructed a static grasping field in 3D space for hand-object interaction based on an RGB image. GraspTTA [17] predicted a static joint-angle configuration of the grasping hand from a given object point cloud and contributed a Test Time Adaptation (TTA) strategy for helping the method generalize to novel objects. Going beyond static grasping synthesis, ManipNet [41] proposed several geometric sensors and managed to generate long-term complex manipulation sequences. TOCH [46] achieves a data-driven approach of dynamic motion refinement in hand object motion synthesis.

3. Problem Formulation and Notations

In this paper, we focus on synthesizing functional manipulation for a specific task goal defined on a known category of articulated or rigid objects (*e.g.* opening a laptop). The input of our system can be split into three parts:

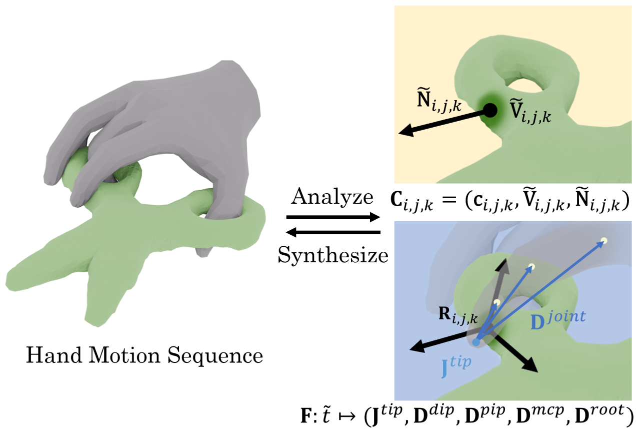
- A **novel object instance** from a known object category $\mathcal{C}$. The object instance is described by triangular object part meshes $\{\mathbf{M}_k\}_{k=1}^N$, where $N > 1$ indicates an articulated object and $N = 1$ indicates a rigid object. We assume that the number of rigid parts N for each object category is known and constant.
- A **goal sequence** $\mathbf{G} = \{(\mathbf{S}^j \rightarrow \mathbf{S}^{j+1}, \mathbf{t}^j)\}_{j=0}^M$ that defines the goal of a manipulation task by the movement of object's parts, in which t^j denotes the lasting time of the j-th stage, and $\mathbf{S}^j = \{\mathbf{S}_k^j \in SE(3)\}_{k=1}^N$ denotes the 6D poses of each rigid part at the j-th stage transition point. The whole manipulation procedure is subdivided into multiple stages according to the object's state of movement (*e.g.* A goal sequence of the *opening* task for laptops consists of two stages: *approaching* and *opening*, as shown in Figure 1).
- An **initial hand pose** $\mathbf{H}_0 \in \mathbb{R}^{51}$ represented by the 48-dimensional MANO [32] pose parameters and a 3D wrist position.

The goal is to generate a sequence of human-like hand motions that is represented by a hand pose sequence $\{\theta_t \in \mathbb{R}^{51}\}_{t=1}^T$. Such a sequence of motion should be consistent with the given goal sequence of the manipulated object. To make this generation task reasonable, the given goal sequence should conform to the object's articulation constraint and follow its functionality.

4. Method

Figure 2 shows the framework of our system. We first introduce **CAnonicalized Manipulation Spaces** (**CAMS**), a two-level space hierarchy that allows representing dynamic hand motions in an object-centric and contact-centric view (see Section 4.1). By embedding hand motions in CAMS, we obtain a more compact description for dynamic manipulations, which is more friendly to learning.

To learn to synthesize human-like manipulations, we propose a two-stage framework consisting of a CVAE-based **planner** module (see Section 4.2) and an optimization-based **synthesizer** module (see Section 4.3). Taking the object shape $\mathbf{O}$, the manipulation goal $\mathbf{G}$ and the initial hand configuration $\mathbf{H}_0$ as input, we first divide the whole manipulation process into several motion stages by the sequential goals. Then, the planner module generates stage-wise **contact targets** and time-continuous **finger embeddings** as guidance for HOM synthesis. Finally, the synthesizer module takes the generated contact targets and hand embeddings



$\tilde{\mathbf{N}}_{i,j,k}$
$\tilde{\mathbf{V}}_{i,j,k}$
Analyze
$\mathbf{C}_{i,j,k} = (\mathbf{c}_{i,j,k}, \tilde{\mathbf{V}}_{i,j,k}, \tilde{\mathbf{N}}_{i,j,k})$
Synthesize
Hand Motion Sequence
$\mathbf{R}_{i,j,k}$
$\mathbf{D}^{joint}$
$\mathbf{J}^{tip}$
$\mathbf{F}: \tilde{t} \mapsto (\mathbf{J}^{tip}, \mathbf{D}^{dip}, \mathbf{D}^{pip}, \mathbf{D}^{mcp}, \mathbf{D}^{root})$

Figure 3. **CAMS Representation of Hand Motion.** We present Canonicalized Manipulation Spaces, a novel representation for hand-object interaction, to express the motion under an object-centric and contact-centric view. The top right figure demonstrates discrete contact targets defined at each stage transition point. The bottom right figure shows the time-continuous finger embedding in one frame. See Section 4.1 for details.

as inputs and leverages an optimization strategy to produce a human-like and physically realistic HOM animation.

4.1. CAnonicalized Manipulation Spaces (CAMS)

It is challenging to model the space of hand manipulation from the given object shape due to the shape variety within a category and the huge diversity of human manipulation styles. For two object instances that have a non-negligible geometry difference, even if we apply the same manipulation style on them (*e.g.* same finger placement on two pairs of scissors with different sizes), the resulting hand pose can also be significantly different. Hence it is difficult for previous approaches that directly learn the MANO parameters to generalize to novel object instances, especially when object shapes are similar during training. To address this challenge, we propose to express the motion of each finger in a canonical reference frame centered at a contact point on the object, thus associating such motion to the local contact region rather than the whole object shape.

Based on this motion expression, we introduce CAnonicalized Manipulation Spaces with two-level canonicalization for manipulation representation. At the root level, the *canonicalized contact targets* (see Section 4.1.1) describe the discrete contact information. At the leaf level, the *canonicalized finger embedding* (see Section 4.1.2) transforms finger motion from global space into local reference frames defined on the contact targets.

4.1.1 Canonicalized Contact Targets

At the root level of CAMS, the canonicalized contact targets describe the contact information between each finger and the object. The contact targets are first used to define the contact-centric reference frames for finger embeddings (Section 4.1.2), and then allow the contact optimization of the synthesizer to achieve grasps with accurate contacts (Section 4.3).

Formally, we define the contact targets as a sequence

$$\mathbf{C} = \{(\mathbf{c}_{i,j,k}, \tilde{\mathbf{V}}_{i,j,k}, \tilde{\mathbf{N}}_{i,j,k})\}_{i,j,k},$$

in which $1 \le i \le 5$ indicates the index of finger, $0 \le j \le M$ indicates the index of stage transition point and $1 \le k \le N$ indicates the index of object's part. $\mathbf{c}_{i,j,k} \in [0,1]$ are binary flags of whether the finger i is in contact with the object part k at stage transition point j, and $\tilde{\mathbf{V}}_{i,j,k} \in \mathbb{R}^3, \tilde{\mathbf{N}}_{i,j,k} \in \mathbb{R}^3$ are canonicalized positions and normal directions of the corresponding contact point. We follow NPCS [22, 36] to normalize all the rigid parts into a normalized coordinate space within the unit cube (i.e., $x, y, z \in [0,1]$), aligned with a category-level canonical orientation. Contact positions $\tilde{\mathbf{V}}_{i,j,k}$ as well as the surface normals $\tilde{\mathbf{N}}_{i,j,k}$ are defined in the normalized space, in order to make the representation more compactly distributed and easier to learn.

Our generative model learns the distribution of discrete contact targets conditioned on the object shape and task configuration. The benifit of learning surface normal $\tilde{\mathbf{N}}_{i,j,k}$ together with position $\tilde{\mathbf{V}}_{i,j,k}$ is two fold. First, for an unseen object with complex shapes (*e.g.* scissors), due to the limitation of network capacity and training data diversity, we cannot guarantee that the predicted contact positions are always exactly on the object surface. When projecting the contact point onto the surface, the predicted normal directions help filter out the incorrect surface parts near the predicted point. Second, the difference in normal directions among different finger placement styles is more significant. It makes it easier for the VAE-based model to distinguish the discrete manipulation styles if normal directions are regarded as reconstruction targets.

At the training stage, the ground truth contact targets are obtained by applying a contact analysis to training data. At the inference stage, given $\tilde{\mathbf{V}}_{i,j,k}$ and $\tilde{\mathbf{N}}_{i,j,k}$ with $\mathbf{c}_{i,j,k} = 1$, the actual contact point $\mathbf{P}_{i,j,k}$ on the object surface is determined by a matching process on the object surface. For more details, please refer to our supplementary material.

4.1.2 Canonicalized Finger Embeddings

Given contact points $\mathbf{P}_{i,j,k}$ from contact targets, we can now build contact-centric spaces for finger embedding. We denote $\mathbf{R}_{i,j,k}$ as the contact reference frame centered at $\mathbf{P}_{i,j,k}$, with orientation aligned to the corresponding rigid part of the object (as shown in 3). For a static hand pose represented by MANO parameters θ and β, we first calculate the corresponding MANO joint coordinates $\mathbf{J}^{tip}, \mathbf{J}^{dip}, \mathbf{J}^{pip}, \mathbf{J}^{mcp}$ of finger i and $\mathbf{J}^{root}$ for the hand

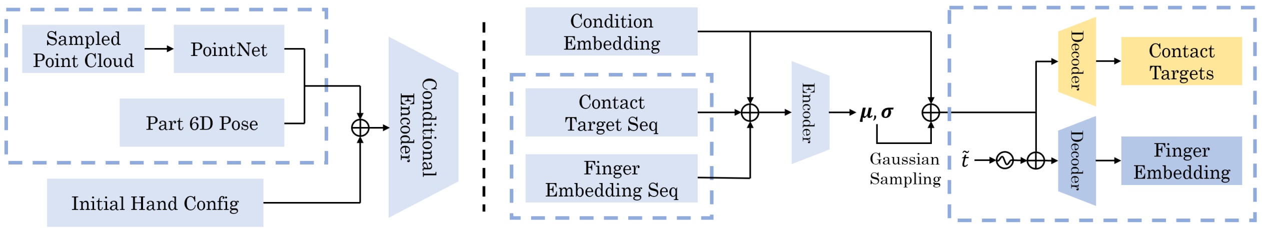


Figure 4. **Network Architecture of CAMS-CVAE.** On the left side, we show the design of the condition encoder, in which the system input is encoded into a condition embedding for CVAE. The components in the dotted box are repeated for each rigid part of the object. On the right side, we show the architecture of the whole CVAE. The components in the dotted boxes are repeated for each tuple (i, j, k).

wrist under reference frame $\mathbf{R}_{i,j,k}$. Based on joint positions, we further compute the normalized directions

$$\mathbf{D}^{joint} = \frac{\mathbf{J}^{joint} - \mathbf{J}^{tip}}{\|\mathbf{J}^{joint} - \mathbf{J}^{tip}\|}, joint \in \{dip, pip, mcp, root\}$$

from the fingertip to every other joint. The resulting finger embedding for the static hand pose is $\mathbf{F} = (\mathbf{J}^{tip}, \mathbf{D}^{dip}, \mathbf{D}^{pip}, \mathbf{D}^{mcp}, \mathbf{D}^{root}) \in \mathbb{R}^{15}$. Similar to contact targets, we aim to enlarge the discrepancy between different modes by using directions as learning targets. Also, directions between joints are invariant under global translation relative to the contact center, and therefore data noise caused by the contact center's fluctuation can be reduced. The tip joint is selected as the reference joint since it's typically close to the contact position in most of the manipulation styles.

To handle dynamic hand motion synthesis, we further extend the static finger embedding to *continuous-time sequences*. More specifically, for a finger interacting with the object at a stage in time period $[t_0, t_1]$, the continuous-time finger embedding sequence is a continuous function $\mathbf{f} : [0, 1] \rightarrow \mathbb{R}^{15}$ that maps the normalized time $\tilde{t} = (t - t_0)/(t_1 - t_0)$ to a finger embedding $\mathbf{F} \in \mathbb{F}^{15}$. Inspired by recent works [27, 29, 31] of implicit neural representation, the continuous mapping is learned by a neural network which maps the *temporal encoding* $\mathcal{T} \in \mathbb{R}^{12}$ defined as

$$\mathcal{T}(\tilde{t}) = \{(\sin 2^k \pi \tilde{t}, \cos 2^k \pi \tilde{t})\}_{k=0}^{5}$$

together with a latent code to $\mathbf{F} \in \mathbb{R}^{15}$ (see Section 4.2).

4.2. CAMS-CVAE: the Motion Planner

Given an object instance with task configuration, we present CAMS-CVAE, a CVAE-based generative motion planner for generating CAMS sequences, including contact targets and continuous-time finger embeddings. An overview of the model architecture is shown in Figure 4.

Condition Encoding As mentioned earlier, our model takes several inputs as generating conditions, including the object part meshes $\{\mathbf{M}_k\}_{k=1}^{N}$, the goal sequence $\mathbf{G}$ and the initial hand configuration $\mathbf{H}_0$. We encode this condition information into a vector using a multi-head condition encoder module, with each head corresponding to an object's rigid part. For each part, we first sample 2000 points on the mesh surface with normal directions, and a PointNet structure is used to embed the point cloud information $\mathbf{O} \in \mathbb{R}^{2000\times6}$ into a shape feature $\boldsymbol{z}_O \in \mathbb{R}^{32}$. The shape feature is then concatenated with the part's 6D-pose sequence $\mathbf{S} \in \mathbb{R}^{6\times(M+1)}$, and all part's information is concatenated together with the initial hand pose $\mathbf{H}_0 \in \mathbb{R}^{51}$. Finally, an MLP structure encodes the concatenated vector into the condition embedding $\boldsymbol{z}_C \in \mathbb{R}^{32}$.

Motion Encoding In the encoder part of the CVAE structure, we encode the hand motion represented by CAMS sequences into a diagonal-covariance Gaussian distribution in a 64-dimensional latent space. We first concatenate the whole sequence of contact targets into a vector $\mathbf{C} \in \mathbb{R}^{7\times5\times N\times(M+1)}$ (as introduced in Section 4.1.1). In each stage, we evenly spaced sample 10 timestamps $\{0, 1/9, 2/9, \cdots, 1\}$ in the normalized time range and calculated the corresponding finger embeddings (as introduced in Section 4.1.2) under contact targets at the stage's two end-points. Besides finger embeddings, we also extract two binary flags $\mathbf{f}_c, \mathbf{f}_n$ (see Section 4.3) used in the synthesizer module at each sampled timestamp for each tuple (i, j, k). For non-existing contact targets with $\mathbf{c}_{i,j,k} = 0$, the corresponding values are filled with zero. All these values are concatenated are encoded by an MLP to predicted Gaussian parameters $\boldsymbol{\mu} \in \mathbb{R}^{64}, \boldsymbol{\sigma} \in \mathbb{R}^{64}$.

Motion Decoding After sampling latent code $\boldsymbol{z} \in \mathbb{R}^{64}$ from either the predicted distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ or standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$, we concatenate it with the generating condition $\boldsymbol{z}_C$, and a two-branch decoder is used to convert them into CAMS representation of desired hand motion. The *discrete branch* is a multi-head MLP that outputs discrete contact targets $(\mathbf{c}_{i,j,k}, \tilde{\mathbf{V}}_{i,j,k}, \tilde{\mathbf{N}}_{i,j,k}) \in \mathbb{R}^7$ introduced in Section 4.1.1, with each head corresponding to a tuple (i, j, k) indicating the index of the finger, stage transition point, and object's part. The *continuous-time branch* is another multi-head MLP that takes the temporal encoding $\mathcal{T}(\tilde{t}) \in \mathbb{R}^{12}$ as extra input and outputs the corresponding finger embeddings $\mathbf{F}_1, \mathbf{F}_2 \in \mathbb{R}^{15}$ relative to contact targets at both end-points of the stage. The continuous-time branch also predicts the two binary flags $\mathbf{f}_c, \mathbf{f}_n$ for the synthesizer.

Training Loss We use several losses to train CAMS-CVAE. The first loss is a Binary Cross Entropy (BCE) loss $\mathcal{L}_{flag}$ on the contact target flags $\mathbf{c}_{i,j,k}$ and per-frame flags $\mathbf{f}_c, \mathbf{f}_n$ for synthesizer, between predicted values and the ground truth. We then calculate the L_2 distance losses $\mathcal{L}_{pos}$ of $\tilde{\mathbf{V}}_{i,j,k}$ and $\mathcal{L}_{dir}$ for $\tilde{\mathbf{N}}_{i,j,k}$, between predicted values and the ground truth. These two losses are computed only for (i, j, k) with the ground truth contact flag $\mathbf{c}_{i,j,k} = 1$. Similarly, we also compute the L_2 losses $\mathcal{L}_{tip}$ for predicted $\mathbf{J}^{tip}$ and $\mathcal{L}_{vec}$ for predicted $\mathbf{D}^{dip}, \mathbf{D}^{pip}, \mathbf{D}^{mcp}, \mathbf{D}^{root}$. Finally, we use the KL-Divergence $\mathcal{L}_{KLD}$ to constrain the latent distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ to be close to the standard Gaussian distribution following [19]. The total loss is a weighted summation of all six loss terms:

$$\begin{aligned}\mathcal{L} = \lambda_{flag} \cdot \mathcal{L}_{flag} + \lambda_{pos} \cdot \mathcal{L}_{pos} + \lambda_{dir} \cdot \mathcal{L}_{dir} \\ + \lambda_{tip} \cdot \mathcal{L}_{tip} + \lambda_{vec} \cdot \mathcal{L}_{vec} + \lambda_{kld} \cdot \mathcal{L}_{KLD}.\end{aligned} \tag{1}$$

For a detailed calculation of the loss terms, please refer to our released code.

4.3. Optimization-Based Motion Synthesizer

Once a CAMS embedding sequence has been generated from the planner, our optimization-based synthesizer subsequently produces a complete HOM sequence. We simply use Bézier curve-based interpolation to generate the object trajectory and thus focus on synthesizing a hand trajectory that satisfies our goal (Section 3). Given the object shape and trajectory, as well as a CAMS embedding sequence, our synthesizer adopts a two-stage optimization method that first optimizes the MANO pose parameters to best fit the CAMS finger embedding (see Section 4.3.1) and then optimizes the contact effect to improve physical plausibility (see Section 4.3.2). In each frame, the synthesizer reads a binary flag $\mathbf{f}_n$ indicating whether the finger is near enough (10cm) that the finger embedding will be used to guide the generated motion, and a binary flag $\mathbf{f}_c$ indicating whether there is a contact between the finger and object.

4.3.1 Fitting Finger Embedding

Given the predicted contact targets $\mathbf{C}$, the bidirectional finger embedding $\mathbf{F}_1, \mathbf{F}_2$ and the binary flags $\mathbf{f}_c, \mathbf{f}_n$, we optimize the MANO parameters of hand pose and transition $\theta = \{\theta_t\}_{t=1}^{T}$ by minimizing:

$$\begin{aligned}\mathcal{L}(\theta) = &\lambda_{\text{tip}} \sum_{t=1}^{T} \mathcal{L}_{\text{tip}}(\theta_t) + \lambda_{\text{joint}} \sum_{t=1}^{T} \mathcal{L}_{\text{joint}}(\theta_t) \\ &+ \lambda_{\text{smooth}} \mathcal{L}_{\text{smooth}}(\theta).\end{aligned} \tag{2}$$

The first term of Eq.(2) is the tip transition loss, which constrains the tip position $\mathbf{J}^{tip}$ of the finger in the canonicalized finger embedding. The second term of Eq.(2) is the joint orientation loss, which is used to optimize the direction vectors of four subsequent joints $\mathbf{D}^{dip}$, $\mathbf{D}^{pip}$, $\mathbf{D}^{mcp}$, $\mathbf{D}^{root}$ of the finger in the canonicalized finger embedding. And the last term of (2) is a smoothness loss for improving temporal continuity.

Category	Task
Laptop	Open the Laptop
Bucket	Lift the Handle
Scissors	Open the Scissors
Pilers	Clamp
Kettle	Pick and Place

Table 1. **Categories and Tasks**. We select five object categories (four articulated object categories and one rigid object category) with different manipulation tasks from HOI4D [25].

4.3.2 Optimizing Contact and Penetration

After fitting MANO parameters θ by finger embedding, we leverage another optimization-based method to handle the penetration and inaccurate contact issues of the hand pose. The optimization course refines θ to physically realistic MANO parameters θ' as the final result of synthesis.

To achieve better contact quality, we define a contact loss $\mathcal{L}_{\text{contact}}$ to attract the nearby finger vertices to the local surface section. And to avoid penetration, we also use a penetration loss $\mathcal{L}_{\text{penetr}}$ by repulsing hand mesh vertices that are inside the object mesh.

Besides, three additional losses $\mathcal{L}_{\text{trans}}(\theta'_{\dots;0,1,2}) = \sum_t \|\theta'_{t;0,1,2} - \theta'_{t+1;0,1,2}\|^2$, $\mathcal{L}_v(\theta') = \|\dot{\mathbf{p}}\|^2$ and $\mathcal{L}_a(\theta') = \|\ddot{\mathbf{p}}\|^2$ are utilized for smoothening the wrist transition, velocity and acceleration of hand joints, respectively.

We minimize the overall loss value defined as

$$\begin{aligned}\mathcal{L}(\theta') = &\lambda_{\text{contact}}(\mathcal{L}_{\text{contact}}(\theta') + \mathcal{L}_{\text{penetr}}(\theta')) \\ &+ \lambda_{\text{trans}} \mathcal{L}_{\text{trans}}(\theta'_{\dots;0,1,2}) \\ &+ \lambda_{\text{smooth}}(\lambda_v \mathcal{L}_v(\theta') + \lambda_a \mathcal{L}_a(\theta')).\end{aligned} \tag{3}$$

To produce more accurate contacts, we iteratively conduct such an optimization process for several epochs. In each epoch we feed the current θ' to the optimization course for further improvement.

5. Experiments

In this section, we apply our method to synthesize HOM, and evaluate the effect of our method with various metrics. We first introduce the experimental settings including dataset (Section 5.1), baselines (Section 5.3), and evaluation metrics (Section 5.2), while the experimental details are presented in our supplementary material. We then show that our method could handle shape diversity and manipulation diversity and achieve a surpassing synthesis performance compared with other methods in Section 5.5.

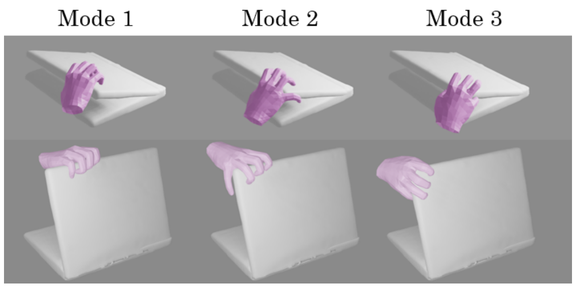


Figure 5. **Visualization of different grasp modes on the same laptop.** We show 3 different grasp modes for opening a laptop. Both the starting and ending hand poses of the grasps are shown.

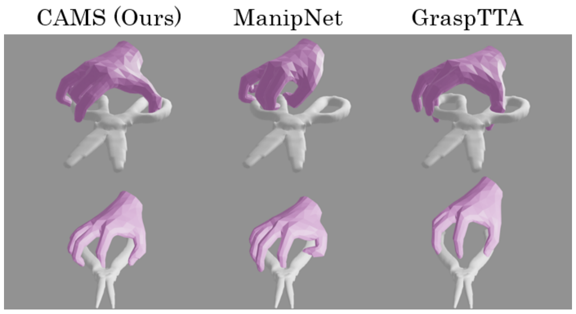


Figure 6. **Qualitative results compared with GraspTTA [17] and ManipNet [41].** The results generated by our method are more realistic.

5.1. Dataset

We utilize HOI4D Dataset [25] in our experiment. HOI4D is a real-world dataset that contains dynamic HOM data spanning various rigid and articulated object categories. We select five object categories with different functionalities in our experiment, as shown in (Tab. 1). The selected HOM data contains both various manipulation types (*e.g*. opening a laptop with different human preferences) and complex manipulation processes (*e.g*. opening a small scissor by putting fingers through the hole) to improve the diversity and complexity of our synthesis results. To improve the usability of HOI4D under our setting, we applied some data cleaning and augmentation methods to the raw data (see our supplementary material for more details).

5.2. Evaluation Metrics

We report several evaluation metrics on our experiments. We use these metrics to quantize the human-likeness and physical plausibility of motion synthesis results.

Contact-Movement Consistency We evaluate whether the object's movement can align with the contact forces produced by hand-object contacts, using the same physics model as in ManipNet [41]. For articulated objects, we assume there are free opposite forces at the spin axis of the object. We calculate the proportion of frames that the contacts align with the object motion.

Articulation Consistency For articulated objects, we evaluate whether the hand pose can manipulate the object in a human-like manner. In particular, for each part of the object in each frame, we compute the torque of all contact points w.r.t. the object's spin axis if a unit force is applied along the normal direction of the contact point. If the maximal attitude of the torque with the same direction of object rotation exceeds a threshold, we regard such frame as qualified. We calculate the proportion of qualified frames.

Penetration Rate We compute the mean penetration proportion of hand vertices for each sequence. We regard penetrations within a small threshold λ=5mm as not penetrated since small penetrations can be seen as contacts made by a soft hand in real.

Perceptual Score We collect human perceptual scores to judge the naturalness of the motion sequences. We ask people not familiar with motion synthesis to give discrete perceptual scores for the results and calculate a mean score for each category using each baseline method. The detailed approach is left to the supplementary material.

5.3. Baselines

As introduced in Section 2, there are only a few works about dynamic HOM generation using a learning-based method, while there are various techniques that could have the potential to be used in this task. Consequently, we design the following baselines.

GraspTTA [17]: GraspTTA proposed a strong baseline in static grasp generation. We use it to generate static grasps of several key snapshots in manipulation and then refine the result at test time using the TTA loss (We do not refine the network at test time). The dynamic HOM animation is thus generated from interpolation based on these snapshots.

ManipNet [41]: Benefiting from carefully designed geometric sensors, ManipNet has shown a strong ability to generalization on generic object manipulation synthesis tasks. Different from our setting, ManipNet assumes additional input of wrist trajectory. To compare it with other methods, we provide it with the wrist trajectory generated by CAMS as input (and thus it only differs at fingers).

Though both GraspTTA and ManipNet have their own mechanisms to improve synthesis quality (*e.g*. reduce penetration), they can also benefit from our optimization-based motion synthesizer (optimizing $\mathcal{L}_{penetr}$ to reduce penetration). We combine all baselines with an optimization stage (denoted as "w/ opt") and compare them with the CAMS-CVAE motion planner.

5.4. Ablation Studies

Remove Contact Optimization: To demonstrate the advantages of contact optimizations in motion synthesis, we train a baseline model with the contact optimizations removed (CAMS-).

	Pliers			Scissors			Laptop		
	Pen (%) ↓	Mov ↑	Art ↑	Pen (%) ↓	Mov ↑	Art ↑	Pen (%) ↓	Mov ↑	Art ↑
Ground Truth	0.000	1.000	1.000	0.046	1.000	0.970	0.316	1.000	1.000
GraspTTA	0.555	0.779	0.420	0.454	0.993	0.849	5.211	**1.000**	0.997
GraspTTA w/ opt	0.294	0.727	0.321	0.812	0.994	0.959	4.702	**1.000**	**1.000**
ManipNet	0.548	0.984	0.892	0.391	0.917	0.417	3.550	**1.000**	0.995
ManipNet w/ opt	0.387	0.890	0.738	0.131	0.831	0.333	2.762	**1.000**	0.994
CAMS-	0.563	0.916	0.393	0.590	0.997	0.850	5.204	**1.000**	0.983
CAMS (Ours)	**0.004**	**1.000**	**1.000**	**0.080**	**0.999**	**0.989**	**0.906**	**1.000**	**1.000**
	Kettle			**Bucket**			**Overall**		
	Pen (%) ↓	Mov ↑	Art ↑	Pen (%) ↓	Mov ↑	Art ↑	Pen (%) ↓	Mov ↑	Art ↑
Ground Truth	0.602	1.000	N/A	0.090	1.000	1.000	0.211	1.000	0.992
GraspTTA	4.852	0.586	N/A	1.309	**1.000**	0.863	2.476	0.872	0.782
GraspTTA w/ opt	3.496	0.642	N/A	1.244	**1.000**	0.886	2.110	0.873	0.791
ManipNet	0.760	0.892	N/A	0.231	**1.000**	0.844	1.096	0.959	0.787
ManipNet w/ opt	0.459	0.622	N/A	0.156	**1.000**	0.865	0.779	0.869	0.733
CAMS-	2.494	0.759	N/A	0.151	**1.000**	0.893	1.800	0.934	0.780
CAMS (Ours)	**0.098**	**0.915**	N/A	**0.015**	**1.000**	**1.000**	**0.221**	**0.983**	**0.997**

Table 2. **Quantitative results compared with GraspTTA [17] and ManipNet [41].** "Pen" denotes the average percentage of hand vertices penetrated in the object. "Mov" denotes the average proportion of frames that are contact-movement consistent. "Art" denotes the average proportion of frames that are articulation consistent (see Section 5.2).

	Perceptual Score $\{1, \ldots, 5\} \uparrow$		
	Pliers	Scissors	Laptop
Ground Truth	4.145	4.025	3.980
GraspTTA	1.243	2.550	1.450
ManipNet	2.000	1.281	2.400
CAMS-	1.986	1.200	1.980
CAMS (Ours)	**3.841**	**3.300**	**3.600**
	Kettle	Bucket	Overall
Ground Truth	3.511	3.920	3.882
GraspTTA	2.146	1.520	1.717
ManipNet	1.988	2.143	2.069
CAMS-	2.202	2.860	2.101
CAMS (Ours)	**2.360**	**3.700**	**3.290**

Table 3. **Perceptual Score.**

Besides removing the contact optimization in the synthesizer, we also did several ablation studies using different representation spaces of fingers. These experiments are left to our supplementary material.

5.5. Comparison

Quantitative Results Table 2 and Table 3 show the quantitative results of our method and all baselines. Our method outperforms previous work on all tasks, even if contact optimization is applied to them.

An observation is that after applying the offline optimization stage, the performance gain of our method is significantly higher than the baselines. This can be explained by that the $\mathcal{L}_{contact}$ term in contact optimization takes the contact targets from our planner as input, and it is the key to producing gradients guiding the finger placement. Without intermediate contact target information, the optimization can only leverage the $\mathcal{L}_{penetr}$ loss term, and it may push the finger out of the object in unpredictable directions.

Qualitative Results Besides Fig. 1, Fig. 5 shows that our method can generate diverse grasp modes on a single object instance. Fig. 6 shows that our method can generate reasonable poses given complex object shapes. We also show full result demonstrations in our video, including the generated whole HOM processes for different tasks, robustness to different object shapes and sizes, comparison between baseline methods, and diversity of generated manipulation styles.

6. Conclusion

In this work, we tackle a novel task of category-level functional hand-object manipulation synthesis. To generate human-like and physically realistic manipulation sequences, we design a two-level space hierarchy named CAnonicalized Manipulation Spaces (CAMS) and thus present a two-stage framework containing a planner and a synthesizer that leverage CAMS as an intermediate representation. Our method achieves state-of-the-art performance for both articulated and rigid object categories.

References

[1] Emre Aksan, Manuel Kaufmann, and Otmar Hilliges. Structured prediction helps 3d human motion modelling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7144–7153, 2019. 2

[2] Emad Barsoum, John Kender, and Zicheng Liu. Hp-gan: Probabilistic 3d human motion prediction via gan. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1418–1427, 2018. 2

[3] Samarth Brahmbhatt, Ankur Handa, James Hays, and Dieter Fox. Contactgrasp: Functional multi-finger grasp synthesis from contact. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2386–2393, 2019. 1

[4] Yujun Cai, Yiwei Wang, Yiheng Zhu, Tat-Jen Cham, Jianfei Cai, Junsong Yuan, Jun Liu, Chuanxia Zheng, Sijie Yan, Henghui Ding, et al. A unified 3d human motion synthesis model via conditional variational auto-encoder. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11645–11655, 2021. 2

[5] Zhe Cao, Ilija Radosavovic, Angjoo Kanazawa, and Jitendra Malik. Reconstructing hand-object interactions in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12417–12426, 2021. 3

[6] Siyu Chen, Alfredo Glioti, Riccardo Rattazzi, Lorenzo Ricci, and Andrea Wulzer. Learning from radiation at a very high energy lepton collider. *Journal of High Energy Physics*, 2022(5):1–58, 2022. 3

[7] Sammy Christen, Muhammed Kocabas, Emre Aksan, Jemin Hwangbo, Jie Song, and Otmar Hilliges. D-grasp: Physically plausible dynamic grasp synthesis for hand-object interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20577–20586, 2022. 1, 2

[8] Enric Corona, Albert Pumarola, Guillem Alenya, and Francesc Moreno-Noguer. Context-aware human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6992–7001, 2020. 2

[9] Qiongjie Cui, Huaijiang Sun, and Fei Yang. Learning dynamic relationships for 3d human motion prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6519–6527, 2020. 2

[10] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David J Crandall. Hope-net: A graph-based model for hand-object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6608–6617, 2020. 3

[11] Carlo Ferrari and John F Canny. Planning optimal grasps. In *ICRA*, volume 3, page 6, 1992. 2

[12] Anand Gopalakrishnan, Ankur Mali, Dan Kifer, Lee Giles, and Alexander G Ororbia. A neural temporal model for human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12116–12125, 2019. 2

[13] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11807–11816, 2019. 3

[14] Gustav Eje Henter, Simon Alexanderson, and Jonas Beskow. Moglow: Probabilistic and controllable motion synthesis using normalising flows. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020. 2

[15] Fangzhou Hong, Mingyuan Zhang, Liang Pan, Zhongang Cai, Lei Yang, and Ziwei Liu. Avatarclip: Zero-shot text-driven generation and animation of 3d avatars. *arXiv preprint arXiv:2205.08535*, 2022. 2

[16] Siyuan Huang, Zan Wang, Puhao Li, Baoxiong Jia, Tengyu Liu, Yixin Zhu, Wei Liang, and Song-Chun Zhu. Diffusion-based generation, optimization, and planning in 3d scenes. *arXiv preprint arXiv:2301.06015*, 2023. 2

[17] Hanwen Jiang, Shaowei Liu, Jiashun Wang, and Xiaolong Wang. Hand-object contact consistency reasoning for human grasps generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11107–11116, 2021. 1, 3, 7, 8

[18] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael Black, Krikamol Muandet, and Siyu Tang. Grasping field: Learning implicit representations for human grasps, 2020. 3

[19] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 6

[20] Jiye Lee and Hanbyul Joo. Locomotion-action-manipulation: Synthesizing human-scene interactions in complex 3d environments. *arXiv preprint arXiv:2301.02667*, 2023. 2

[21] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13401–13412, 2021. 2

[22] Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3706–3715, 2020. 4

[23] Min Liu, Zherong Pan, Kai Xu, Kanishka Ganguly, and Dinesh Manocha. Deep differentiable grasp planner for high-dof grippers. *arXiv preprint arXiv:2002.01530*, 2020. 2

[24] Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. Semi-supervised 3d hand-object poses estimation with interactions in time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14687–14697, 2021. 3

[25] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21013–21022, 2022. 2, 6, 7

[26] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. Weakly-supervised action transition learning for stochastic human motion prediction. In *Proceedings of the IEEE/CVF Con-*

ference on Computer Vision and Pattern Recognition, pages 8151–8160, 2022. 2

[27] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 5

[28] Andrew T Miller and Peter K Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine*, 11(4):110–122, 2004. 2

[29] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 5

[30] Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10985–10995, 2021. 2

[31] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 5

[32] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610*, 2022. 3

[33] Qijin She, Ruizhen Hu, Juzhan Xu, Min Liu, Kai Xu, and Hui Huang. Learning high-dof reaching-and-grasping via dynamic representation of gripper-object interaction. *arXiv preprint arXiv:2204.13998*, 2022. 2

[34] 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)*, 2020. 3

[35] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+ o: Unified egocentric recognition of 3d hand-object poses and interactions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4511–4520, 2019. 3

[36] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019. 4

[37] Jiashun Wang, Huazhe Xu, Jingwei Xu, Sifei Liu, and Xiaolong Wang. Synthesizing long-term 3d human motion and interaction in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9401–9411, 2021. 2

[38] Yan Wu, Jiahao Wang, Yan Zhang, Siwei Zhang, Otmar Hilliges, Fisher Yu, and Siyu Tang. Saga: Stochastic whole-body grasping with contact. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VI*, pages 257–274. Springer, 2022. 2

[39] Zeshi Yang, Kangkang Yin, and Libin Liu. Learning to use chopsticks in diverse gripping styles. *ACM Transactions on Graphics (TOG)*, 41(4):1–17, 2022. 2

[40] Yuting Ye and C Karen Liu. Synthesis of detailed hand manipulations using contact sampling. *ACM Transactions on Graphics (ToG)*, 31(4):1–10, 2012. 2

[41] He Zhang, Yuting Ye, Takaaki Shiratori, and Taku Komura. Manipnet: Neural manipulation synthesis with a hand-object spatial representation. *ACM Transactions on Graphics (ToG)*, 40(4):1–14, 2021. 1, 3, 7, 8

[42] Xiaohan Zhang, Bharat Lal Bhatnagar, Sebastian Starke, Vladimir Guzov, and Gerard Pons-Moll. Couch: towards controllable human-chair interactions. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V*, pages 518–535. Springer, 2022. 2

[43] Yan Zhang, Michael J Black, and Siyu Tang. We are more than our joints: Predicting how 3d bodies move. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3372–3382, 2021. 2

[44] Yan Zhang and Siyu Tang. The wanderings of odysseus in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20481–20491, 2022. 2

[45] Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv preprint arXiv:1703.10960*, 2017. 2

[46] Keyang Zhou, Bharat Bhatnagar, Jan Eric Lenssen, and Gerard Pons-Moll. Toch: Spatio-temporal object correspondence to hand for motion refinement. *arXiv preprint arXiv:2205.07982*, 2022. 3