

# Hand-Object Interaction Controller (HOIC): Deep Reinforcement Learning for Reconstructing Interactions with Physics

Haoyu Hu

hythu17@163.com

School of Software and BNRIst,  
Tsinghua University  
Beijing, China

Xinyu Yi

yixy20@mails.tsinghua.edu.cn  
School of Software and BNRIst,  
Tsinghua University  
Beijing, China

Zhe Cao

zhecao@google.com  
Google

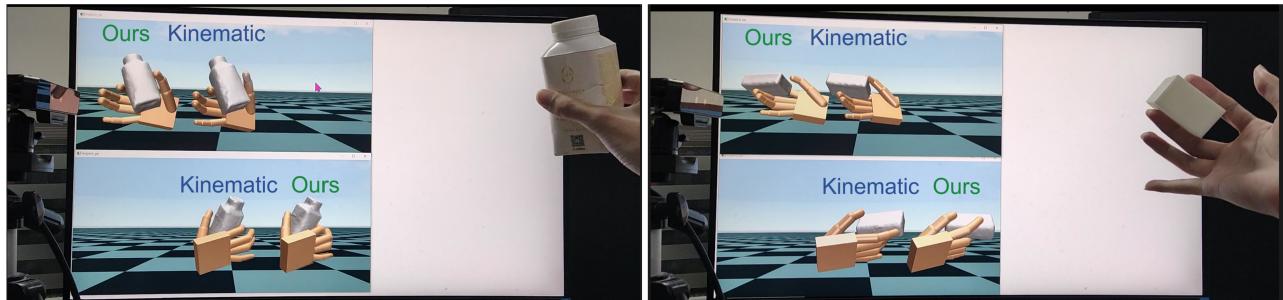
Seattle, United States of America

Jun-Hai Yong

yongjh@tsinghua.edu.cn  
School of Software and BNRIst,  
Tsinghua University  
Beijing, China

Feng Xu\*

xufeng2003@gmail.com  
School of Software and BNRIst,  
Tsinghua University  
Beijing, China



**Figure 1:** Our HOIC framework reconstructs accurate and physically plausible hand-object interaction motions by imitating the vision-based kinematic tracking results in the physics simulator.

## ABSTRACT

Hand manipulating objects is an important interaction motion in our daily activities. We faithfully reconstruct this motion with a single RGBD camera by a novel deep reinforcement learning method to leverage physics. Firstly, we propose object compensation control which establishes direct object control to make the network training more stable. Meanwhile, by leveraging the compensation force and torque, we seamlessly upgrade the simple point contact model to a more physical-plausible surface contact model, further improving the reconstruction accuracy and physical correctness. Experiments indicate that without involving any heuristic physical rules, this work still successfully involves physics in the reconstruction of hand-object interactions which are complex motions hard to imitate with deep reinforcement learning. Our code and data are available at <https://github.com/hu-hy17/HOIC>.

\*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

*SIGGRAPH Conference Papers '24, July 27-August 1, 2024, Denver, CO, USA*

© 2024 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0525-0/24/07  
<https://doi.org/10.1145/3641519.3657505>

## CCS CONCEPTS

- Computing methodologies → Motion capture.

## KEYWORDS

Hand Tracking, Hand-Object Interaction, Single Depth Camera, Deep Reinforcement Learning, Physical Simulation

## ACM Reference Format:

Haoyu Hu, Xinyu Yi, Zhe Cao, Jun-Hai Yong, and Feng Xu. 2024. Hand-Object Interaction Controller (HOIC): Deep Reinforcement Learning for Reconstructing Interactions with Physics. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24 (SIGGRAPH Conference Papers '24), July 27-August 1, 2024, Denver, CO, USA*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3641519.3657505>

## 1 INTRODUCTION

Our hands play a crucial role in our interactions with the world around us. Whether we're shopping and selecting items from shelves or preparing food by skillfully wielding a knife to cut ingredients into small pieces, our hands are at the forefront of these actions. The accurate and real-time reconstruction of such Hand-Object Interaction (HOI) motions can facilitate a variety of applications including virtual reality, human-computer interaction, and robot learning.

Many works have focused on the challenging task of jointly reconstructing hand pose and object motion from monocular RGB

or depth input only. To deal with the ill-posed nature of this problem, some recent studies utilize physical laws to regularize or refine the reconstruction procedures, which can leverage both data-driven and model-driven methodologies to yield more accurate and physically plausible results [Hu et al. 2022; Tzionas et al. 2016; Wang et al. 2013; Zhao et al. 2022]. However, these methods either perform time-consuming optimization processes that are not suitable for real-time applications [Tzionas et al. 2016; Wang et al. 2013; Zhao et al. 2022] or exert a strong restriction in the scheme when performing the physical refinement [Hu et al. 2022]. A more powerful solution to incorporate physics in the reconstruction of HOI is eagerly required.

Imitation learning techniques are proven to be successful in generating physically plausible results and have been widely used in human motion synthesis [Peng et al. 2018, 2021; Yuan and Kitani 2020] and capture [Fussell et al. 2021; Yuan et al. 2021] in recent years. These techniques usually train an imitation policy network under the reinforcement learning framework. The policy network takes reference non-physical motions as input and output control signals to generate physically plausible motions. However, applying these techniques in the field of HOI reconstruction still faces some challenges: Firstly, a HOI controller needs to accomplish a more complex task than that of a human body controller. Under HOI scenarios, the objects can only be indirectly controlled through the contact force exerted by the hand. The HOI controller not only has to mimic the reference hand motion but also needs to generate accurate contact points and forces to drive the manipulated object and ensure the object follows the reference object motion, which is very challenging to learn. Secondly, the contact mechanism in modern physics simulators differs significantly from the HOI contact in the real physical world. In the real world, human hands have soft tissues that can deform and generate contact areas when contacting an object. In traditional simulators, however, an approximation is made using contact points. This limitation prevents many real interactive actions from being imitated in the physical simulation.

In this paper, we propose HOIC (Hand-object Interaction Controller), a deep reinforcement learning (DRL) method for imitating and reconstructing HOI motion, which effectively incorporates physics and addresses the aforementioned challenges. We introduce an innovative mechanism called object compensation control, designed to address the discrepancies between the physical simulator’s contact representation and the real-world contact mechanism. In addition to hand control signals, our policy network generates supplementary force and torque, directly influencing the object during simulation. This direct control significantly improves system stability in the imitation task. The forces and torques, referred to as compensation force and torque, could be well explained by a surface contact model. This model implicitly allows the system to simulate surface contacts instead of point contacts during hand-object interactions. In this manner, we avoid the task of modeling the soft tissue surface contact in the forward physical model, which is known as a complex and time-consuming process. Finally, the unexplained compensation force and torque are formed as the penalty of a reward term during policy training. The proposed object compensation control not only simplifies the HOI imitation task but also enhances its physical plausibility. The approach ensures a more direct and realistic interaction between the hand and the object, contributing to the overall effectiveness of the imitation process.

In the experiments, we demonstrate that object compensation control can significantly improve training speed and imitation quality. We also incorporate the imitation policy into a real-time HOI tracking system and prove that it can reconstruct more physically plausible HOI motions compared with the pure vision-based method [Zhang et al. 2021b] and the method with heuristic physical refinement [Hu et al. 2022].

## 2 RELATED WORKS

As our work focuses on reconstructing physically plausible HOI motion via the motion imitation controller, we will review recent works on HOI reconstruction and physics-aware human motion control.

### 2.1 Hand-object Interaction Reconstruction

Joint reconstruction of interacting hand object motions has been studied for many years. The heavy occlusion between hand and object during interaction makes this task challenging and ill-posed. Earlier works use multi-view camera input to minimize the impact of occlusion [Ballan et al. 2012; Oikonomidis et al. 2011]. With the development of depth cameras, some works [Chen et al. 2023c; Kyriazis and Argyros 2013; Schmidt et al. 2015; Sridhar et al. 2016; Tsoli and Argyros 2018; Zhang et al. 2021b] utilize RGBD images as input to obtain more 3D cues, improving reconstruction precision. In recent years, many HOI datasets [Brahmbhatt et al. 2020; Chao et al. 2021; Fan et al. 2023; Garcia-Hernando et al. 2018; Hampali et al. 2020] have been released, which encourage the community to leverage data-driven methods to reduce the ambiguity of hand object pose estimation [Chen et al. 2023a; Hasson et al. 2019; Karunaratnakul et al. 2020; Tekin et al. 2019].

However, these works still suffer from some noticeable, especially physically implausible artifacts including jitter, penetration, and unstable manipulate pose. Some studies introduce physical priors to regularize the estimated hand object pose. Wang et al. [2013] leverage physical simulation and trajectory optimization to find an optimal motion control that best matches the input data. Tzionas et al. [2016] and Zhao et al. [2022] optimize hand pose in each frame to form a stable object grasping. However, the optimization approaches used in these works are either offline or time-consuming, which hinders their usage in real-time applications. Hu et al. [2022] utilizes object dynamics and a simplified frictional contact model to correct hand pose to match object movement, which can run in real-time. However, it assumes that contact only happens on fingertips, thus being unable to handle general HOI scenarios. Our method overcomes these shortcomings by leveraging the imitation learning technique where the physical prior is learned by mimicking various interaction motions.

### 2.2 Physics-Aware Human Motion Control

**2.2.1 Optimization-based Method.** Many works on human motion incorporate physical laws using optimization-based methods. Some works achieve physically plausible body motion capture by optimizing joint torques and foot contact forces to drive dynamic simulation [Shimada et al. 2020; Xie et al. 2021; Yi et al. 2022; Zheng and Yamane 2013]. In the realm of HOI, the dynamics of hands and objects, and the physical mechanisms of contact are studied and

widely applied in motion synthesis tasks [Kry and Pai 2006; Liu 2009; Mordatch et al. 2012; Wu et al. 2023; Ye and Liu 2012]. For example, Liu [2009] generates various motions in response to different dynamic situations during manipulation by leveraging the physical relationship between object motion and hand actuation torques. Wu et al. [2023] utilizes a bilevel optimization to synthesize diverse grasping motions on different objects while maintaining physical correctness. However, due to the complex contact mechanisms in HOI, these optimization methods are either time-consuming or designed only for grasping tasks. Therefore, our work aims to construct a more efficient and general physical optimization method in HOI using DRL.

**2.2.2 DRL-based Method.** In recent years, deep reinforcement learning (DRL) has emerged as a powerful approach for controlling physically simulated characters, yielding impressive results. Peng et al. [2018] learns a control policy by imitating example data, which can generate high-quality, physically plausible human motions. Yuan and Kitani [2020] improves the ability of the mimic policy by introducing residual force control. Instead of imitating a single motion at once, some works [Peng et al. 2022, 2021; Tessler et al. 2023] utilize GAIL [Ho and Ermon 2016] to learn stylized motion skills from multiple motion clips, which can generate diverse movements by combining different skills. Besides motion synthesis, some works utilize DRL for motion capture. Yuan et al. [2021] trains an imitation control policy to follow the reference human motion captured by the vision-based method. Winkler et al. [2022] uses DRL to learn the mapping between sparse signals from VR devices and full-body motion. By incorporating modern physical simulators, these methods can yield physically plausible results when reconstructing real-world human motion.

DRL is also widely used in HOI control for robot learning and motion synthesis. Andrychowicz et al. [2020] and Chen et al. [2022b] train hand controllers to finish object reorientation tasks on robotic hands. Chen et al. [2022a] explores several manipulation tasks involving two dexterous hands. Some works use DRL to learn complex interaction skills, such as using chopsticks [Yang et al. 2022], playing basketball [Liu and Hodgins 2018], and pre-grasping [Chen et al. 2023b]. Imitation learning has gained popularity in recent years, aiming to enhance sample efficiency during training and the naturalness of manipulation motions yielded by the policy. Sermanet et al. [2018] and Liu et al. [2018] improve the manipulation ability of robots by imitating expert motions extracted from raw videos. However, these works employ two-finger grippers which enable fewer interactive movements compared to the human hand. Rajeswaran et al. [2017], Radosavovic et al. [2021], and Qin et al. [2022] perform imitation learning using a dexterous hand model, generating manipulation motions that are close to the real human hand. However, all these works focus on finishing a specific manipulation task such as moving, pouring, reorientation, etc. Our goal is to develop a more general policy that can be used to imitate any given human-object interaction.

### 3 METHOD

We aim to reconstruct hand-object interaction motion from a single-view camera in real-time. Our method takes a monocular RGBD

video stream as input and outputs the hand pose and the object motion of each frame (Fig. 2). Firstly, we use an off-the-shelf kinematic tracking method [Zhang et al. 2021b] to obtain a coarse estimation of hand pose  $\tilde{\mathbf{q}}_t$  and object pose  $\tilde{\mathbf{o}}_t$ . Without careful physics-aware modeling, the kinematic interaction motion may contain many physically implausible artifacts such as penetration, jitter, missing contact points, and non-physical sliding. Then the kinematic results are used as reference motions and an imitation process is performed to refine the kinematic results (Sec. 3.1). To be specific, at each time step  $t$  of the imitation, a policy  $\pi$  takes the refined result  $\{\mathbf{q}_t, \mathbf{o}_t\}$  at  $t$  and several future frames of the kinematic results  $\{\tilde{\mathbf{q}}_{t+1}, \tilde{\mathbf{o}}_{t+1}, \dots, \tilde{\mathbf{q}}_{t+N}, \tilde{\mathbf{o}}_{t+N}\}$  as input. The policy outputs control signals, including not only the joint torques  $\tau_t$  for the hand but also the compensation force  $f_t^c$  and torque  $\tau_t^c$  for the object, which are all fed into a physical simulator to simulate physical-plausible results. To train the policy, we utilize two rewards: the mimic reward  $r_t^{\text{mimic}}$ , and the physics reward  $r_t^{\text{phys}}$ .

The mimic reward is a standard imitation learning metric that encourages the policy to replicate the reference motion. The physics reward acts as a constraint, ensuring that the training is stable while the results adhere to physical laws. More specifically, to compute  $r_t^{\text{phys}}$ , we use a surface contact model to explain the compensation force (force and torque, using force for simplification) for more physical correctness (Sec. 3.2). While the surface contact model can explain most of the compensation force, a portion remains unaccounted for, which is referred to as the residual force. The physics reward guides the policy towards physically realistic compensation force usage by minimizing the residual force. This ensures that the generated forces align with the laws of physics.

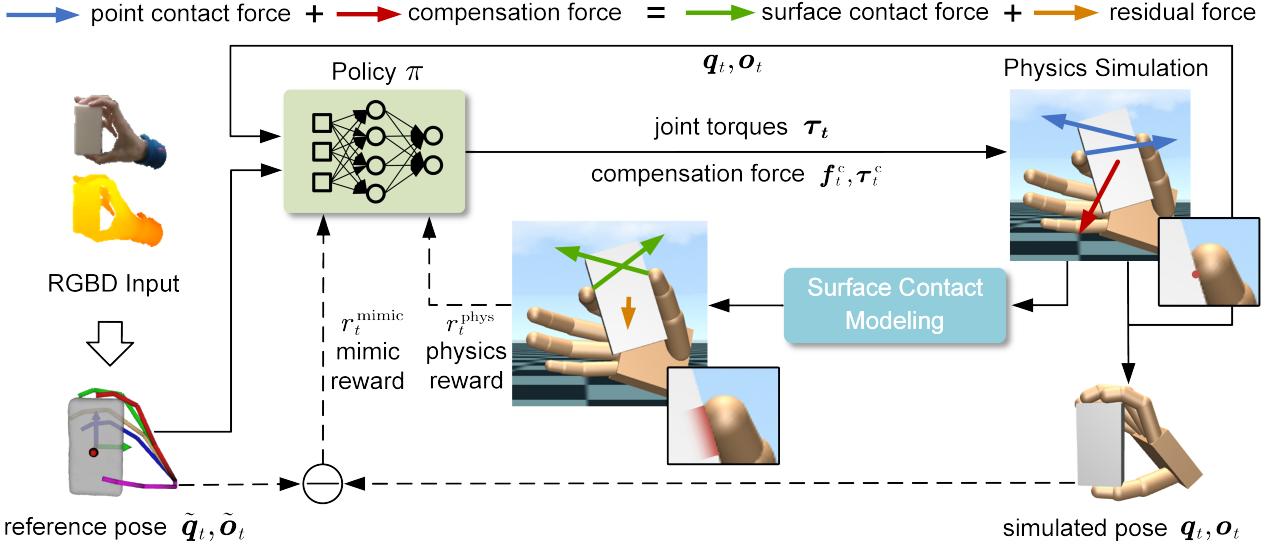
### 3.1 HOI Imitation Learning

To imitate the hand-object interaction with a physics simulator, we apply a standard reinforcement learning framework [Sutton et al. 1998]. Reinforcement learning aims to find an optimal solution for a sequential decision process. In the process, an agent interacts with the environment according to a policy  $\pi$  which is typically represented as a conditional distribution. At each time step of the sequence, the agent observes the state  $s_t$  of the environment and takes an action  $a_t$  sampled from the distribution  $\pi(a|s)$ . After taking the action, the agent receives a reward  $r(s_t, a_t)$  and the new state  $s_{t+1}$ . The optimization objective is to find the optimal distribution  $\pi(a|s)$  that maximizes the accumulated rewards over the entire sequence, written as:

$$\arg \max_{\pi} E_{\tau \sim p_{\pi}(\tau)} \left( \sum_t r(s_t, a_t) \right), \quad (1)$$

where  $\tau$  stands for the whole state and action sequence  $[s_0, a_0, s_1, a_1, \dots, s_T, a_T]$ , and  $p_{\pi}(\tau)$  is the distribution of  $\tau$  when the agent follows the policy  $\pi$ .

Deep Reinforcement Learning (DRL) uses a neural network to represent the policy, enabling the learning of complex mappings between high-dimensional environmental states and actions. In our work, we follow the widely used reinforcement learning algorithm Proximal Policy Optimization (PPO) [Schulman et al. 2017] to train our policy network. In the following part of this section,



**Figure 2: System Overview.** Our system takes the kinematic tracking results of hand-object interaction  $\{\tilde{q}_t, \tilde{o}_t\}$  as input and outputs a refined version  $\{q_t, o_t\}$ . The policy network  $\pi$  first takes  $\{q_t, o_t\}$  and  $\{\tilde{q}_{t+i}, \tilde{o}_{t+i}\}$  as input to predict the control signals  $\{\tau_t, f_t^c, \tau_t^c\}$  which are fed into a physical simulator to obtain  $\{q_{t+1}, o_{t+1}\}$ . In the training process, the compensation force  $\{f_t^c, \tau_t^c\}$  is applied to upgrade the contact model from point to surface contact in a Surface Contact Modeling step and the residual is used to construct the physics reward  $r_t^{\text{phys}}$ . Meanwhile, a mimic reward  $r_t^{\text{mimic}}$  is also used to train the policy network  $\pi$ .

we will elaborate on the definition of state, action, and reward in our method.

**3.1.1 State.** The state is defined as:

$$s_t = (q_t, \dot{q}_t, o_t, \dot{o}_t, \tilde{q}_{t+i}, \tilde{o}_{t+i}), \quad i = 1 \dots N, \quad (2)$$

which contains not only the hand pose  $q_t$ , the hand velocity  $\dot{q}_t$ , the object pose  $o_t$ , and the object velocity  $\dot{o}_t$  at the current frame, but also the kinematic hand pose  $\tilde{q}_{t+i}$  and the kinematic object pose  $\tilde{o}_{t+i}$  in the future frames. Considering that the action should be independent of the absolute positions of the hands and objects, we do not include the current hand root translation in the state and transform all the components in Eq. 2, except hand root rotation contained in  $q_t$ , to the current hand root coordinate.

**3.1.2 Action.** The action  $a_t$  contains control signals that are used to drive the proxy hand and object in the physics simulator. For hand control, instead of directly outputting joint torques from the policy, we first output a target hand pose  $\hat{q}_t$  and employ a PD-controller to solve for the joint torques following previous approaches [Christen et al. 2022; Yuan et al. 2021]. To be specific, the PD-controller takes the current hand pose  $q_t$ , velocity  $\dot{q}_t$ , and the target hand pose  $\hat{q}_t$  as input. It outputs joint torques  $\tau_t$  that drive the hand to move towards the target pose, written as:

$$\tau_t = k_p \circ (\hat{q}_t - q_t) - k_d \circ \dot{q}_t, \quad (3)$$

where  $\circ$  denotes vector element-wise multiplication,  $k_p$  and  $k_d$  are the parameters of the PD-controller.

Different from previous works on hand manipulation control, we augment our action space with a compensation force  $f_t^c, \tau_t^c$  that

is directly exerted on the object. Physically, the object is driven by the hand and the compensation force does not exist. However, in the reinforcement learning framework, it is difficult to train the policy for the hand joints with rewards defined on the manipulated object, as the control is quite indirect. Involving this compensation force makes the training more stable and thus enables our policy to track a wider range of interaction motions.

Finally, our action is defined as:  $a_t = \{\tau_t, f_t^c, \tau_t^c\}$ . Note that the problem of the physical incorrectness involved by the compensation force will be handled later.

**3.1.3 Reward.** The goal of our policy is to imitate a given hand-object interaction motion while using a physically valid compensation force to control the object. Therefore, the reward at each time step is defined as the multiplication of two types of reward: mimic reward  $r_t^{\text{mimic}}$  and physics reward  $r_t^{\text{phys}}$ :

$$r_t = r_t^{\text{mimic}} \cdot r_t^{\text{phys}}. \quad (4)$$

The mimic reward  $r_t^{\text{mimic}}$  encourages the policy to generate a similar interaction motion to the reference motion, which is defined as:

$$r_t^{\text{mimic}} = r_t^{\text{hand}} \cdot r_t^{\text{object}}, \quad (5)$$

where  $r_t^{\text{hand}}$  evaluates the similarity of hand motion between the imitation result and the reference. Inspired by previous work on human body imitation [Yuan et al. 2021], our hand reward  $r_t^{\text{hand}}$  consists of four parts:

$$r_t^{\text{hand}} = w^{\text{pose}} r_t^{\text{pose}} + w^{\text{joint}} r_t^{\text{joint}} + w^{\text{orient}} r_t^{\text{orient}} + w^{\text{vel}} r_t^{\text{vel}}, \quad (6)$$

where  $r_t^{\text{pose}}$  measures the difference between the current hand pose  $\mathbf{q}_t$  and the reference hand pose  $\tilde{\mathbf{q}}_t$ :

$$r_t^{\text{pose}} = \exp(-k^{\text{pose}}\|\mathbf{q}_t - \tilde{\mathbf{q}}_t\|). \quad (7)$$

$r_t^{\text{joint}}$  evaluates the difference between the current 3D hand joint positions  $\mathbf{j}_t$  and the reference hand joint positions  $\tilde{\mathbf{j}}_t$ :

$$r_t^{\text{joint}} = \exp(-k^{\text{joint}}\|\mathbf{j}_t - \tilde{\mathbf{j}}_t\|). \quad (8)$$

$r_t^{\text{orient}}$  encourages the orientation of each bone  $R_t^i$  of the current hand is close to that of the reference  $\tilde{R}_t^i$ , computed as:

$$r_t^{\text{orient}} = \exp\left(-k^{\text{orient}} \sum_{i=1}^B |R_t^i \ominus \tilde{R}_t^i|\right), \quad (9)$$

where  $B$  is the number of bones and  $\ominus$  denotes the relative rotation angle between the two rotation matrices.  $r_t^{\text{vel}}$  guarantees the velocity of hand motion  $\dot{\mathbf{q}}_t$  should be similar to the reference motion  $\dot{\tilde{\mathbf{q}}}_t$ :

$$r_t^{\text{vel}} = \exp(-k^{\text{vel}}\|\dot{\mathbf{q}}_t - \dot{\tilde{\mathbf{q}}}_t\|), \quad (10)$$

where  $\dot{\mathbf{q}}_t$  comes from the simulator whereas  $\dot{\tilde{\mathbf{q}}}_t$  is calculated using finite difference. Similar to  $r_t^{\text{hand}}$ ,  $r_t^{\text{object}}$  measures the object motion difference, which is computed as:

$$r_t^{\text{object}} = w^{\text{opose}} r_t^{\text{opose}} + w^{\text{oavel}} r_t^{\text{oavel}}, \quad (11)$$

where  $r_t^{\text{opose}}$  and  $r_t^{\text{oavel}}$  are calculated similar to  $r_t^{\text{pose}}$  in Eq. 7 and  $r_t^{\text{vel}}$  in Eq. 10, respectively.

Different from previous imitation learning methods [Peng et al. 2018; Yuan et al. 2021], our method incorporates a novel physics reward  $r_t^{\text{phys}}$  in addition to the imitation reward. This physics reward is designed to constrain the use of compensation forces. While compensation forces enable the policy to imitate a wider range of hand-object interactions, they can also be exploited to maximize imitation rewards in ways that violate physical laws. To prevent this, we aim to ensure that the policy’s compensation force remains within a physically plausible range. A naive solution is to restrict the compensation force to be as small as possible as it is not a real existing force. However, we notice that the point contact model used in the physics simulator is efficient but not physically correct. The compensation force could be used to model this incorrectness and thus does not need to be zero. Based on this observation, we propose to use a surface contact model (detailed in Sec. 3.2) to measure how much of the compensation force is modeling the surface contact and we just restrict the residue, so-called residual force, to be as small as possible. In this manner, the physics reward  $r_t^{\text{phys}}$  reflects how well the simulation adheres to real physics.

Note that all the sub-rewards, including the imitation rewards for both the hand and the object, as well as the physics reward, are multiplied together to form the final reward. This design forces the policy to accurately imitate hand and object movements while employing compensation forces in a physically realistic manner.

### 3.2 Surface Contact Modeling

In this section, we will elaborate on the computation of physics reward  $r_t^{\text{phys}}$  using our surface contact model. For brevity, the subscript for time step  $t$  will be omitted in the remainder of this section.

The input of surface contact modeling consists of hand-object contact points  $\{\mathbf{p}_i\}$ , normals  $\{\mathbf{n}_i\}$ , object mass center  $\mathbf{c}$ , translation acceleration  $\dot{\mathbf{v}}$ , angular velocity  $\boldsymbol{\omega}$  and acceleration  $\dot{\boldsymbol{\omega}}$ , mass  $m$ , and inertia  $\mathcal{I}$ . These data can be obtained from the physics simulator. Firstly, we expand each contact point to form a contact surface. Some previous works implement surface contact mechanisms by using a deformable hand model [Jain and Liu 2011] or by adding torsional and rolling torques to the point contact model [Luding 2008]. However, the former needs to use the meshes of hand and object to compute an accurate contact surface, which is undesirable considering the high sample efficiency required by reinforcement learning. The latter needs additional physical hyperparameters to ensure accuracy. Therefore, we consider a simple but efficient model in our method. For each contact point, we sample four additional points alongside four tangential directions, and these points are 2.5 millimeters away from the original point. These additional points, combined with the original contact points, are referred to as extended contact points. It can be proved that on the rectangle enclosed by the additional points, any distribution of surface contact forces can be equivalently represented by a combination of point contact forces at the extended contact points [Caron et al. 2015]. After the expansion, we obtained a set of new contact points that is five times more than the original ones, denoted as  $\{\mathbf{p}_i^{\text{ext}}\}$ .

Next, we optimize a set of forces at each extended contact point to match the dynamics of the object. Similar to the previous work [Hu et al. 2022], we use the Coulomb Friction Model to represent the contact force. The contact force is composed of pressure and frictional force, and the resultant force of these two must be within the range of the friction cone. Following [Lynch and Park 2017], we use a quadrangular pyramid to approximate the friction cone for computational convenience. Each contact force  $\mathbf{f}_i$  is represented by the positive span of four normalized lateral edges  $\mathbf{x}_1, \dots, \mathbf{x}_4$  of the pyramid, written as  $\mathbf{f}_i = \mathbf{A}_i \boldsymbol{\lambda}_i$ , ( $\boldsymbol{\lambda}_i \in \mathbb{R}^4, \boldsymbol{\lambda}_i \geq 0$ ), where  $\mathbf{A}_i = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4]$ . The optimization problem is defined as

$$\begin{aligned} E(\boldsymbol{\lambda}_i) &= E_{\text{force}} + E_{\text{torque}} + E_{\text{rel}} \\ \text{s.t. } \boldsymbol{\lambda}_i &\geq 0. \end{aligned} \quad (12)$$

$E_{\text{force}}$  requires that the sum of contact forces and the object gravity  $m\mathbf{g}$  match the translation acceleration  $\dot{\mathbf{v}}$  of the object:

$$\begin{aligned} E_{\text{force}} &= \left\| \sum_{i=1}^P \mathbf{A}_i \boldsymbol{\lambda}_i - \mathbf{f}_{\text{target}} \right\|^2, \\ \mathbf{f}_{\text{target}} &= m\dot{\mathbf{v}} - m\mathbf{g}, \end{aligned} \quad (13)$$

where  $P$  is the number of extended contact points. Similar to  $E_{\text{force}}$ ,  $E_{\text{torque}}$  evaluates the consistency between total contact torques and the rotational movement of the object:

$$\begin{aligned} E_{\text{torque}} &= \left\| \sum_{i=1}^P [\mathbf{p}_i^{\text{ext}} - \mathbf{c}] \times \mathbf{A}_i \boldsymbol{\lambda}_i - \boldsymbol{\tau}_{\text{target}} \right\|^2, \\ \boldsymbol{\tau}_{\text{target}} &= \mathcal{I}\dot{\boldsymbol{\omega}} + [\boldsymbol{\omega}] \times \mathcal{I}\boldsymbol{\omega}, \end{aligned} \quad (14)$$

where  $\mathbf{c}$  is the mass center of the object,  $[\cdot] \times$  is the cross product matrix. Finally,  $E_{\text{rel}}$  requires that when there is relative sliding between the hand and the object at the contact point, the frictional

force should act in the direction that prevents sliding:

$$E_{\text{rel}} = \sum_{i=1}^P (\beta_i \mathbf{e} + \mathbf{A}_i^T \mathbf{v}_{t_i}^{\text{rel}}) \cdot \lambda_i, \quad (15)$$

$$\beta_i = \|\mathbf{A}_i^T \mathbf{v}_{t_i}^{\text{rel}}\|_\infty,$$

where  $\mathbf{e}$  is a vector of ones,  $\mathbf{v}_{t_i}^{\text{rel}}$  is the relative tangential velocity at the contact point  $\mathbf{p}_i^{\text{ext}}$ ,  $\cdot$  is the dot product of two vectors,  $\|\cdot\|_\infty$  is the maximum among the absolute values of the vector elements. For more details about the meaning of  $E_{\text{rel}}$ , we recommend readers refer to [Andrews et al. 2022].

The optimization problem in Eq. 12 can be solved using quadratic programming. This process yields a solution comprising a set of contact forces, named net contact forces, which optimally align with the dynamics of the object. Note that we just calculate these forces to model the surface contact but have not used them in the imitation process. In our system, we do not need to directly involve them in the imitation process, which is complex and time-consuming. The reason is that we have a compensation force that can be used to represent these forces and is already involved in the imitation process. In this case, the residue in the compensation force, which cannot be represented by the net contact forces, can be treated as the difference between the net contact forces and the proper driving force for object motion, which is named residual force  $f_{\text{res}}$ ,  $\tau_{\text{res}}$ . When the compensation force is correctly used by the policy, the motion of the object matches the contact status between the hand and the object, and the residual force approaches zero. At this point, the interaction motion should be physically reasonable. Therefore, our physics reward  $r^{\text{phys}}$  is computed as:

$$r^{\text{phys}} = \exp \left( -k^{\text{phys}} \left( \|f_{\text{res}}\| + w^{\text{torque}} \|\tau_{\text{res}}\| \right) \right). \quad (16)$$

## 4 EXPERIMENTS

In this section, we first introduce our implementation details and then compare our method with the previous state-of-the-art methods for HOI reconstruction. Next, we evaluate the key system design of our method. Finally, we discuss the limitations of our work. More video results can be found in our supplementary video.

### 4.1 Implementation Details

We utilize SingleDepth [Zhang et al. 2021b] as our kinematic estimator, which is the state-of-the-art method for real-time HOI reconstruction. We use MuJoCo [Todorov et al. 2012] as our physics simulator. Our physical hand model is shown in Fig. 3 (left), which has 26 DoFs and consists of capsules and boxes to represent the phalanges and the palm, respectively. To obtain the object collision model (Fig. 3 (middle)), we first perform a convex decomposition algorithm [Wei et al. 2022] on the object mesh (Fig. 3 (left)) reconstructed by our kinematic estimator. Then we manually adjust the scale of the collision model to better approximate the reconstructed object mesh. To achieve stable simulation, we utilize a frequency of 450Hz in the physics simulator, while the reference sequence operates at a frequency of 30Hz. This means that the simulator will step 15 times after the policy takes action once.

To train the imitation policy, we collect a dataset using our kinematic estimator, which includes various hand-object interaction

**Table 1: Average reward of the last 500 training epochs (the policy nearly converges) under different future frame number settings. Note that 1.0 is the maximum average reward that can be received by the policy.**

Number of future frames ( $N$ )	1	5	10
Average Reward	0.910	<b>0.924</b>	0.920

motions. The dataset includes three objects: box, bottle, and banana, each with approximately 40 minutes (72K frames) of sequences. For each object, we train a policy on the corresponding dataset. During training, we follow the random start and early stop strategies proposed by [Peng et al. 2018]. In each episode, a motion clip is sampled from the dataset and the policy needs to imitate from a random start point till the sequence ends or a significant deviation occurs between the imitated results and the reference. The training procedure takes about 2 days on a 16-core AMD EPYC 9654 CPU and an NVIDIA 4090 GPU, which consists of 200 million samples. The inference speed of the policy is 8ms per frame, which enables us to combine it with our kinematic estimator without affecting the real-time performance. Finally, our full tracking system can run at a speed of 25FPS with about 300ms delay. The delay is majorly caused by using future kinematic results for the policy network  $\pi$ . The number of future frames is controlled by  $N$  in Eq. 2. In practice, we set  $N = 5$ , which means that the policy can take action with the knowledge of five future frames. Taking too many future frames does not significantly improve the policy (as shown in Tab. 1) and, at the same time, increases the system latency. Thus we set  $N = 5$  to make a trade-off.

### 4.2 Comparisons

In the literature, the work of [Zhang et al. 2021b] and [Hu et al. 2022] are the state-of-the-art techniques of jointly tracking hand-object interactions in real-time with a single-view RGBD camera. The former is used as our kinematic estimator while the latter, similar to us, aims to perform real-time physical optimization to better reconstruct the HOI motions. To both evaluate the accuracy and the physical plausibility of the reconstructed interaction motions, we use the following metrics:

- **Average Pixel Error (APE)** Following the previous work [Hu et al. 2022; Zhang et al. 2021b], we project the five fingertips of the tracked hand onto the input color image and evaluate the average pixel error to the ground truth.
- **Object IoU** To evaluate object tracking accuracy, we similarly project objects onto the image and measure the intersection over union (IoU) compared with the ground truth.
- **Physically Plausible Ratio (Phys.)** We evaluate the degree of consistency between the dynamic of the object and the hand-object contact status. This is done with the same optimization approach as described in Sec. 3.2. When the sum of the residual force and torque after optimization is less than 0.01, we consider the frame to be physically plausible.
- **Penetration** To compute the penetration depth, we use collision models in the physics simulator to calculate the contact points between the hand and object for each frame,

**Table 2: Comparison results of tracking accuracy (APE, Object IoU) and physical plausibility (Phys., Pen., Hand Smo., Obj Smo.) with [Zhang et al. 2021b] and [Hu et al. 2022] on three interaction sequences. The unit of penetration is mm while the unit of hand-object smoothness is cm/s<sup>2</sup>. The non-zero penetration value of our method is due to the small difference between the collision object model and the reconstructed object model.**

	Method	[Zhang et al. 2021]	[Hu et al. 2022]	Ours
Box	APE ↓	<b>12.75</b>	13.02	13.09
	Obj IoU ↑	76.25	74.93	<b>78.14</b>
	Phys. ↑	59.56	75.66	<b>91.06</b>
	Pen. ↓	2.25	2.33	<b>1.57</b>
	Hand Smo. ↓	87.86	97.99	<b>70.28</b>
	Obj Smo. ↓	85.84	88.03	<b>70.72</b>
Bottle	APE ↓	12.18	13.50	<b>11.88</b>
	Obj IoU ↑	80.94	80.19	<b>81.12</b>
	Phys. ↑	73.12	95.72	<b>98.74</b>
	Pen. ↓	1.97	1.83	<b>0.225</b>
	Hand Smo. ↓	131.3	139.6	<b>111.7</b>
	Obj Smo. ↓	129.6	129.3	<b>110.1</b>
Banana	APE ↓	10.53	11.34	<b>10.27</b>
	Obj IoU ↑	70.01	73.26	<b>73.53</b>
	Phys. ↑	85.86	91.50	<b>95.30</b>
	Pen. ↓	1.39	1.63	<b>0.870</b>
	Hand Smo. ↓	68.28	79.29	<b>56.82</b>
	Obj Smo. ↓	69.67	68.29	<b>50.38</b>

and then substitute these contact point positions into the SDF function of the reconstructed object mesh.

- **Hand and Object Smoothness (Hand Smo. & Obj Smo.)**  
Similar to [Zhang et al. 2021a], we measure the smoothness of hand motion by the acceleration of the hand’s joints and fingertips, and the smoothness of object motion by the acceleration of the object’s center of mass.

Our test set contains three labeled sequences with 5,417 frames in total, and each sequence corresponds to one object. Tab. 2 shows the quantitative results of the three test sequences. The results show that our method can achieve comparable tracking accuracy compared with our kinematic estimator [Zhang et al. 2021b] while significantly improving physical plausibility compared to both [Zhang et al. 2021b] and [Hu et al. 2022]. Besides, the interaction motions reconstructed by our method are more smooth and with less penetration. Fig. 4 illustrates some qualitative comparison results. The first three rows in the figure show that our method, similar to [Hu et al. 2022], can refine the missing contact points (R1, R2) and the wrong contact position caused by non-physical sliding (R3). However, as [Hu et al. 2022] only models point contacts at the fingertips, it may erroneously optimize some contact points that do not exist (R4, R5), while our method avoids these artifacts by learning from motions with diverse contact status using a surface contact model. Besides, our method exhibits higher robustness when facing rapid interactive movements (R6, R7). We believe this is due to the ability of physical controllers to perform smooth motions. Finally, leveraging the non-penetration nature of physical simulation, our method

**Table 3: Quantitative comparison of our surface contact model (SCM) with point contact model (PCM) and no contact model to explain compensation force (w/o CM). The evaluation is performed on the Box sequence.**

	w/o CM	PCM	SCM (Ours)
APE ↓	13.57	13.38	<b>13.09</b>
Obj IoU ↑	75.85	77.80	<b>78.14</b>
Phys. ↑	73.37	90.16	<b>91.06</b>

effectively handles non-physical phenomena of hand-object penetration that occur in some kinematic tracking results (R8). For more results, please refer to our video.

### 4.3 Evaluation

Firstly, we evaluate the effectiveness of the object compensation control. Fig. 5 shows the training curve of our reinforcement learning framework w/ and w/o compensation control. The result shows that the use of compensation control can significantly ease the learning process and improve the imitation quality. We also demonstrate in the video that without the compensation control, the policy can only imitate some very simple interactive actions (such as static grasping), and cannot imitate more complex actions correctly.

Then, we evaluate the efficacy of our surface contact model (SCM) by replacing it with two other designs: (1) completely remove the surface contact model and simply require minor compensation forces to form the physics reward  $r_t^{\text{phys}}$  (w/o CM), (2) remove the extended contact points used to model surface contact and use a point contact model to compute physics reward (PCM). We compare the performance of these three approaches on the Box sequence. The quantitative results of reconstruction accuracy and physical correctness are shown in Tab. 3, and the qualitative results are shown in Fig. 6. The results indicate that when there is no contact model to explain the compensation force, the policy fails to learn to use the compensation force correctly, and thus more non-physical outcomes occur. Although the point contact model provides a suitable approximation in most cases, it still falls short of accurately representing the actual contact mechanisms in some challenging interaction motions. In contrast, our surface contact model is closer to real hand-object interactions, therefore achieving better results. For more results, please refer to our video.

### 4.4 Limitations and Discussions

The generalization capability of our system is not satisfying for the current technique as the system only works on each of the trained objects. We believe this is due to the complexity of the HOI motion, which makes it very hard to learn the physics laws by the current deep reinforcement learning framework given the inaccurate physical simulator. However, we still believe this is a step further in learning the physics of the complex HOI motion, and its benefits in reconstruction are well demonstrated. From the numerical errors, the tracking accuracy of the hand and object judging from the observed view is not improved largely as the kinematic tracking method already achieves high performance on these metrics by directly optimizing them. On the other hand, we

see our method largely improves the physical plausibility while still maintaining comparable accuracy, which still indicates the advances of our technique. Our technique is only evaluated with rigid objects with simple shapes. It is still an open problem to handle more complex object shapes and motions.

## 5 CONCLUSIONS

We propose HOIC which learns the physics of hand-object interaction motions and improves the reconstruction of these motions in the case with limited recording (i.e. single-view RGBD recording). To the best of our knowledge, this is the first work that enables deep reinforcement learning to learn the physics of complex hand-object interactions. The success of training the HOIC is attributed to a compensation force modeling method, which eases the learning process by directly manipulating the object, rather than using the hand joint torques to indirectly control the object. Furthermore, the non-real compensation force is also able to bridge the difference between the used point contact physical model and the real surface contact. In this manner, the physical simulator does not need to model the soft hand tissues to generate surface contact, which is complex and time-consuming. Experiments demonstrated that our method successfully learns the complex physics of interaction motions and generates more physically plausible reconstruction results with limited observations.

## ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China (2023YFC3305600, 2018YFA0704000), the NSFC (No.62021002), and the Key Research and Development Project of Tibet Autonomous Region (XZ202101ZY0019G). This work was also supported by THUIBCS, Tsinghua University, and BLBCI, Beijing Municipal Education Commission.

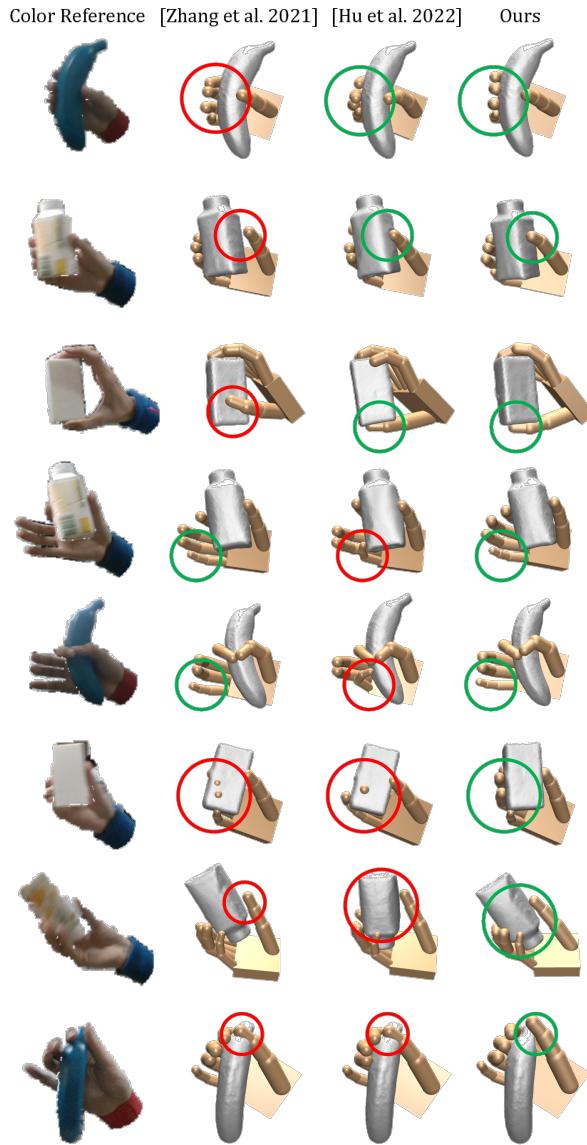
## REFERENCES

- Sheldon Andrews, Kenny Erleben, and Zachary Ferguson. 2022. Contact and friction simulation for computer graphics. In *ACM SIGGRAPH 2022 Courses*. 1–172.
- OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. 2020. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research* 39, 1 (2020), 3–20.
- Luca Ballan, Aparna Tanuja, Jürgen Gall, Luc Van Gool, and Marc Pollefeys. 2012. Motion capture of hands in action using discriminative salient points. In *European Conference on Computer Vision*. Springer, 640–653.
- Samarth Brahmbhatt, Chengcheng Tang, Christopher D. Twigg, Charles C. Kemp, and James Hays. 2020. ContactPose: A Dataset of Grasps with Object Contact and Hand Pose. In *The European Conference on Computer Vision (ECCV)*.
- Stéphane Caron, Quang-Cuong Pham, and Yoshihiko Nakamura. 2015. Stability of surface contacts for humanoid robots: Closed-form formulae of the contact wrench cone for rectangular support areas. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5107–5112.
- 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. 2021. DexYCB: A Benchmark for Capturing Hand Grasping of Objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Jiayi Chen, Mi Yan, Jiazhao Zhang, Yinzhen Xu, Xiaolong Li, Yijia Weng, Li Yi, Shuran Song, and He Wang. 2023c. Tracking and reconstructing hand object interactions from point cloud sequences in the wild. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 304–312.
- Sirui Chen, Albert Wu, and C Karen Liu. 2023b. Synthesizing Dexterous Nonprehensile Pregrasp for Ungraspable Objects. In *ACM SIGGRAPH 2023 Conference Proceedings*. 1–10.
- Tao Chen, Jie Xu, and Pulkit Agrawal. 2022b. A system for general in-hand object re-orientation. In *Conference on Robot Learning*. PMLR, 297–307.
- Yuanpei Chen, Tianhao Wu, Shengjie Wang, Xidong Feng, Jiechuan Jiang, Zongqing Lu, Stephen McAleer, Hao Dong, Song-Chun Zhu, and Yaodong Yang. 2022a. Towards human-level bimanual dexterous manipulation with reinforcement learning. *Advances in Neural Information Processing Systems* 35 (2022), 5150–5163.
- Zerui Chen, Shizhe Chen, Cordelia Schmid, and Ivan Laptev. 2023a. gSDF: Geometry-Driven Signed Distance Functions for 3D Hand-Object Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12890–12900.
- Sammy Christen, Muhammed Kocabas, Emre Aksan, Jemin Hwangbo, Jie Song, and Otmar Hilliges. 2022. D-Grasp: Physically Plausible Dynamic Grasp Synthesis for Hand-Object Interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20577–20586.
- Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J. Black, and Otmar Hilliges. 2023. ARCTIC: A Dataset for Dexterous Bimanual Hand-Object Manipulation. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Levi Fussell, Kevin Bergamin, and Daniel Holden. 2021. Supertrack: Motion tracking for physically simulated characters using supervised learning. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–13.
- Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. 2018. First-Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*.
- Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. 2020. HOnnote: A method for 3D Annotation of Hand and Object Poses. In *CVPR*.
- Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. 2019. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11807–11816.
- Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. *Advances in neural information processing systems* 29 (2016).
- Haoyu Hu, Xinyu Yi, Hao Zhang, Jun-Hai Yong, and Feng Xu. 2022. Physical interaction: Reconstructing hand-object interactions with physics. In *SIGGRAPH Asia 2022 Conference Papers*. 1–9.
- Sumit Jain and C Karen Liu. 2011. Controlling physics-based characters using soft contacts. In *Proceedings of the 2011 SIGGRAPH Asia Conference*. 1–10.
- Korrawe Karunaratana, Jinlong Yang, Yan Zhang, Michael J Black, Krikamol Muandet, and Siyu Tang. 2020. Grasping field: Learning implicit representations for human grasps. In *2020 International Conference on 3D Vision (3DV)*. IEEE, 333–344.
- Paul G Kry and Dinesh K Pai. 2006. Interaction capture and synthesis. *ACM Transactions on Graphics (TOG)* 25, 3 (2006), 872–880.
- Nikolaos Kyriazis and Antonis Argyros. 2013. Physically plausible 3d scene : The single actor hypothesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9–16.
- C Karen Liu. 2009. Dexterous manipulation from a grasping pose. In *ACM SIGGRAPH 2009 papers*. 1–6.
- Libin Liu and Jessica Hodgins. 2018. Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–14.
- YuXuan Liu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. 2018. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 1118–1125.
- Stefan Luding. 2008. Cohesive, frictional powders: contact models for tension. *Granular matter* 10, 4 (2008), 235–246.
- Kevin M Lynch and Frank C Park. 2017. *Modern robotics*. Cambridge University Press.
- Igor Mordatch, Zoran Popović, and Emanuel Todorov. 2012. Contact-invariant optimization for hand manipulation. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*. 137–144.
- Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. 2011. Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *2011 International Conference on Computer Vision*. IEEE, 2088–2095.
- Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. 2018. Deep-mimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)* 37, 4 (2018), 1–14.
- Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. 2022. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions On Graphics (TOG)* 41, 4 (2022), 1–17.
- Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. 2021. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–20.
- Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. 2022. Dexmv: Imitation learning for dexterous manipulation from human videos. In *European Conference on Computer Vision*. Springer, 570–587.
- Ilija Radosavovic, Xiaolong Wang, Lerrel Pinto, and Jitendra Malik. 2021. State-only imitation learning for dexterous manipulation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 7865–7871.
- Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. 2017. Learning complex dexterous

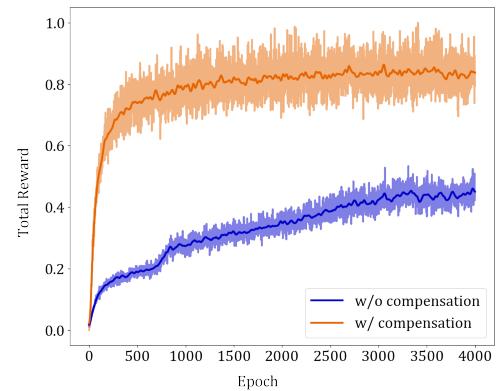
- manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087* (2017).
- Tanner Schmidt, Katharina Hertkorn, Richard Newcombe, Zoltan Marton, Michael Suppa, and Dieter Fox. 2015. Depth-based tracking with physical constraints for robot manipulation. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 119–126.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).
- Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. 2018. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 1134–1141.
- Soshi Shimada, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt. 2020. PhysCap: physically plausible monocular 3D motion capture in real time. *ACM Transactions on Graphics* 39 (dec 2020).
- Srinath Sridhar, Franziska Mueller, Michael Zollhöfer, Dan Casas, Antti Oulasvirta, and Christian Theobalt. 2016. Real-time joint tracking of a hand manipulating an object from rgbd input. In *European Conference on Computer Vision*. Springer, 294–310.
- Richard S Sutton, Andrew G Barto, et al. 1998. *Introduction to reinforcement learning*. Vol. 135. MIT press Cambridge.
- Bugra Tekin, Federica Bogo, and Marc Pollefeys. 2019. 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*, 4511–4520.
- Chen Tessler, Yoni Kasten, Yunrong Guo, Shie Mannor, Gal Chechik, and Xue Bin Peng. 2023. Calm: Conditional adversarial latent models for directable virtual characters. In *ACM SIGGRAPH 2023 Conference Proceedings*. 1–9.
- Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 5026–5033.
- Aggeliki Tsoli and Antonis A Argyros. 2018. Joint 3D tracking of a deformable object in interaction with a hand. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 484–500.
- Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. 2016. Capturing hands in action using discriminative salient points and physics simulation. *International Journal of Computer Vision* 118, 2 (2016), 172–193.
- Yangang Wang, Jianyuan Min, Jianjie Zhang, Yebin Liu, Feng Xu, Qionghai Dai, and Jinxiang Chai. 2013. Video-based hand manipulation capture through composite motion control. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–14.
- Xinyue Wei, Minghua Liu, Zhan Ling, and Hao Su. 2022. Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–18.
- Alexander Winkler, Jungdam Won, and Yuting Ye. 2022. QuestSim: Human motion tracking from sparse sensors with simulated avatars. In *SIGGRAPH Asia 2022 Conference Papers*. 1–8.
- Albert Wu, Michelle Guo, and Karen Liu. 2023. Learning Diverse and Physically Feasible Dexterous Grasps with Generative Model and Bilevel Optimization. In *Conference on Robot Learning*. PMLR, 1938–1948.
- Kevin Xie, Tingwu Wang, Umar Iqbal, Yunrong Guo, Sanja Fidler, and Florian Shkurti. 2021. Physics-based human motion estimation and synthesis from videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 11532–11541.
- Zeshi Yang, Kangkang Yin, and Libin Liu. 2022. Learning to use chopsticks in diverse gripping styles. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–17.
- Yuting Ye and C Karen Liu. 2012. Synthesis of detailed hand manipulations using contact sampling. *ACM Transactions on Graphics (ToG)* 31, 4 (2012), 1–10.
- Xinyu Yi, Yuxiao Zhou, Marc Habermann, Soshi Shimada, Vladislav Golyanik, Christian Theobalt, and Feng Xu. 2022. Physical Inertial Poser (PIP): Physics-aware Real-time Human Motion Tracking from Sparse Inertial Sensors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Ye Yuan and Kris Kitani. 2020. Residual force control for agile human behavior imitation and extended motion synthesis. *Advances in Neural Information Processing Systems* 33 (2020), 21763–21774.
- Ye Yuan, Shih-En Wei, Tomas Simon, Kris Kitani, and Jason Saragih. 2021. Simpoe: Simulated character control for 3d human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 7159–7169.
- He Zhang, Yuting Ye, Takaaki Shiratori, and Taku Komura. 2021a. Manipnet: neural manipulation synthesis with a hand-object spatial representation. *ACM Transactions on Graphics (ToG)* 40, 4 (2021), 1–14.
- Hao Zhang, Yuxiao Zhou, Yifei Tian, Jun-Hai Yong, and Feng Xu. 2021b. Single Depth View Based Real-Time Reconstruction of Hand-Object Interactions. *ACM Transactions on Graphics (TOG)* 40, 3 (2021), 1–12.
- Zimeng Zhao, Binghui Zuo, Wei Xie, and Yangang Wang. 2022. Stability-driven contact reconstruction from monocular color images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1643–1653.
- Yu Zheng and Katsu Yamane. 2013. Human motion tracking control with strict contact force constraints for floating-base humanoid robots. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE, 34–41.



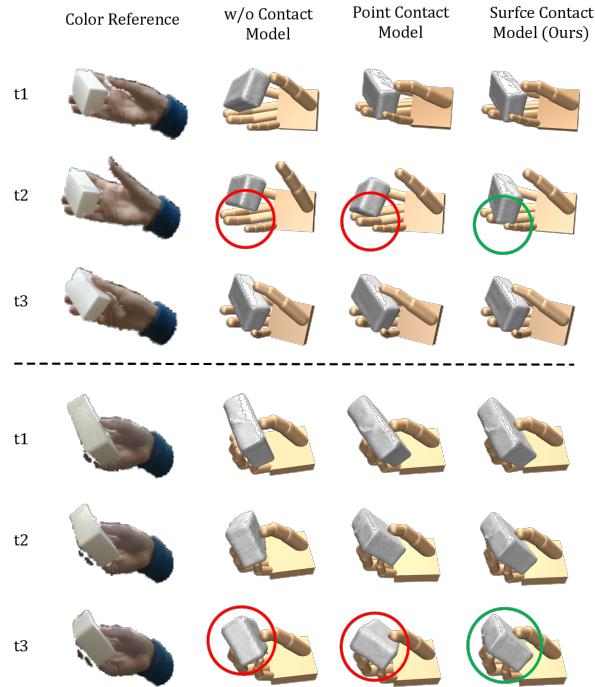
**Figure 3:** Our physical hand model (Left), object mesh reconstructed from kinematic tracking (Middle), and convex object collision mesh (Right).



**Figure 4:** Qualitative comparison with [Zhang et al. 2021b] and [Hu et al. 2022]



**Figure 5:** Training curve of our method with (orange) and without (blue) object compensation control. The Y-axis represents the total rewards obtained by the policy in all episodes within one epoch. The results are normalized by the maximum and minimum values of the total rewards across all epochs.



**Figure 6:** Qualitative evaluation of our surface contact model compared with two other designs: simply requiring minor compensation forces (w/o Contact Model), and a point contact model. Our surface contact model offers more precise modeling of hand-object interaction, particularly in challenging scenarios involving rapidly changing contact status (Top). Additionally, it incorporates torsional and rolling friction torque, enabling more precise imitation of the rotational motion of objects (Bottom).