MS-MANO: Enabling Hand Pose Tracking with Biomechanical Constraints

Pengfei Xie[1,*], Wenqiang Xu[2,*], Tutian Tang[2], Zhenjun Yu[2], Cewu Lu[2]
[1]Southeast University [2]Shanghai Jiao Tong University
[1]xiepf2002@gmail.com [2]{vinjohn, tttang, jeffson-yu, lucewu}@sjtu.edu.cn
https://ms-mano.robotflow.ai

Abstract

This work proposes a novel learning framework for visual hand dynamics analysis that takes into account the physiological aspects of hand motion. The existing models, which are simplified joint-actuated systems, often produce unnatural motions. To address this, we integrate a musculoskeletal system with a learnable parametric hand model, MANO, to create a new model, MS-MANO. This model emulates the dynamics of muscles and tendons to drive the skeletal system, imposing physiologically realistic constraints on the resulting torque trajectories. We further propose a simulation-in-the-loop pose refinement framework, BioPR, that refines the initial estimated pose through a multi-layer perceptron (MLP) network. Our evaluation of the accuracy of MS-MANO and the efficacy of the BioPR is conducted in two separate parts. The accuracy of MS-MANO is compared with MyoSuite, while the efficacy of BioPR is benchmarked against two large-scale public datasets and two recent state-of-the-art methods. The results demonstrate that our approach consistently improves the baseline methods both quantitatively and qualitatively.

1. Introduction

From a physical perspective, human hand motion is actuated by the musculoskeletal system. As depicted in Fig. 1, the brain transmits excitation signals via the nervous system, intriguing the contraction and relaxation in muscles and generating torque to facilitate joint movement of hands. Consequently, the dynamics of hand motion are naturally coordinated and constrained by the underlying musculoskeletal system. However, such physiological aspects are seldom taken into consideration when designing a learning framework of visual hand dynamics analysis (*e.g.* hand pose estimation [1, 28, 46] and tracking [6, 10]).

Previous works on visual hand analysis primarily considered hand dynamics as multi-body dynamics. This means that the hand is represented as an articulated object, with kinematic movement directly propelled by joint torques. Since the joint-actuated system simplifies the mechanism of hand motion, it may produce robot-like movements that are unnatural or infeasible to human hands [14]. In contrast, a musculoskeletal system explicitly emulates the dynamics of muscles and tendons to drive the skeletal system so that it can impose physiologically realistic constraints on the resulting torque trajectories and make the movements more human-like. Despite the advantages, this system is challenging to replicate due to the complexity of the hand's dynamic system, which uses over 30 muscles to support nuanced movements. In this work, we first integrate a musculoskeletal system with a parametric hand model, MANO [35], extending it to the musculoskeletal version, **MS-MANO**. Then, we apply MS-MANO to the hand pose tracking task with a simulation-in-the-loop learning framework, **BioPR**.

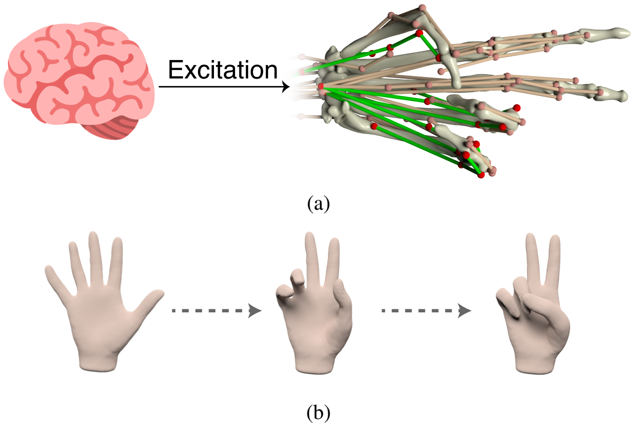


Figure 1. The physiological mechanism of hand dynamics. **(a)** The excitation signal originating from the brain triggers the contraction or relaxation of muscles. The triggered muscle segments are illustrated in green, while the relaxed ones are in brown. **(b)** The muscle contraction triggered by excitation manifests as the movement of the hand in appearance.

[*] Equal contribution.

To build MS-MANO, we focus on three key features: anatomical accuracy, support for learning tasks, and adaptivity to body shape variations. With accurate modeling, it can support precise control of subtle movements and achieves human-like motion; with the support of learning tasks, it can be integrated seamlessly into learning frameworks; with adaptivity, it can accommodate individuals with diverse body shapes. To meet all these goals, we take the muscle-tendon data from MyoHand model [2], which is built upon two anatomic data from OpenSim [39]: MoBL model [30, 36] and the 2nd-Hand models [18]. Then, we integrate this muscle-tendon data into the MANO model [35], a widely recognized and adaptable hand model that can adjust to various hand shapes through different parameters. We aim to have the muscle-tendon structure adaptable to a variety of shapes as well. To realize this, the bone-centric muscle representation is transformed into a joint-centric representation. The details are discussed in Sec. 3.2.

To show the utility of MS-MANO in visual learning tasks, we take the hand pose tracking as our experimental platform. Typically, spatial-temporal features are extracted from observed images. However, when the hand in the image is occluded or motion-blurred, these features may consequently be affected, leading to inconsistent prediction. With biomechanical constraints provided by MS-MANO, we can largely mitigate such instability. To leverage MS-MANO in the hand pose tracking task, we propose a biomechanical pose refinement framework, **BioPR**. BioPR takes the predicted hand pose and velocity (which is inferred from multi-frame poses) as input. It first predicts the excitation signals for all the muscles in the hand model and then uses a simulator to run the hand motion with the excitation to get a reference pose. Next, BioPR refines the initial estimated pose by taking the estimated pose and the reference pose into a multi-layer perception (MLP) network.

The evaluation is conducted in two separate parts: the accuracy of MS-MANO and the efficacy of BioPR. To evaluate the accuracy of MS-MANO, we compare it with MyoSuite by calculating the difference in the trajectories generated. To evaluate BioPR, we adopt two large-scale public datasets that support hand pose tracking: DexYCB [5] and OakInk [49] as benchmarks. Two recent state-of-the-art methods gSDF [6] and Deformer [10] are selected as the baseline methods. With BioPR, the baseline methods are consistently improved quantitatively and qualitatively.

We summarize our contributions as follows:

- We present a musculoskeletal MANO (MS-MANO) hand model. It inherits all the merits of MANO, such as support of learning tasks and adaptivity to different body shapes, but also extends it with musculoskeletal modeling, which can ensure the biomechanical constraints for hand learning tasks.
- We exhibit the ability of MS-MANO in the hand pose tracking task with a simulation-in-the-loop framework, BioPR. We compare the performance of our method with multiple baseline methods on two different benchmarks.

2. Related Works

2.1. Hand Dynamics System

The dynamics of hand motion can be modeled in two distinct ways: through multi-body dynamics and biomechanics. The former conceptualize the human hand as an articulated object, actuating hand movement by directly generating joint torques and may include biomechanical constraints [3]. On the other hand, the latter approach creates musculoskeletal models that utilize biomimetic muscles and tendons to propel skeletal motion. Unlike joint-actuation models, muscle-actuation leads to movements that adhere to physiological constraints [17], and display energy expenditure more akin to actual humans [44]. In the realm of computer animation, the control of a muscle-based virtual character has been investigated in relation to upper body movements [19, 21, 22], hand manipulation [41, 43], and locomotion [11, 20, 23, 31, 40, 44].

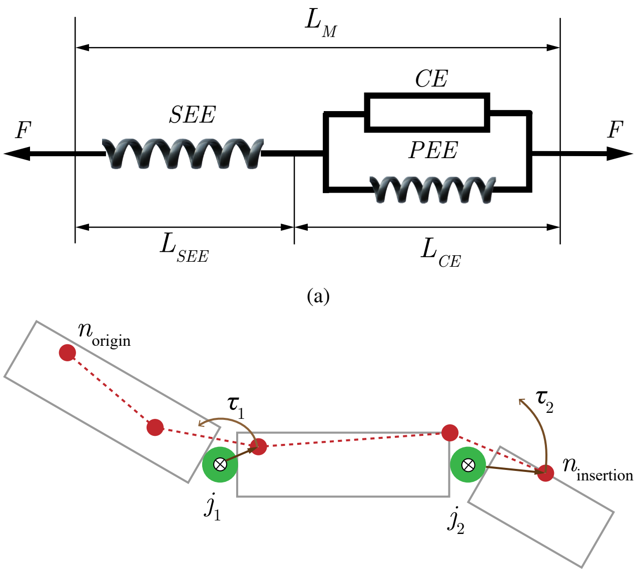
In recent years, the development of statistical parametric human body models has emerged [26, 33, 35]. A series of works have aimed to incorporate the musculoskeletal system into such models, enabling the musculoskeletal structure to adapt according to the parameters. BASH [37] integrates a musculoskeleton into the SMPL model. However, it lacks a complete muscle for the hand. Meanwhile, Ye et al. [52] model a full-body musculoskeletal system, integrating it into the SMPLX model and modeling the mobility-limited behaviors for care recipients. However, these full-body systems do not meticulously model the hand muscle.

In biomechanics, hand movements are not just controlled by the hand but also by the muscles in the forearm. Therefore, a thorough musculoskeletal model for the hand should also include the forearm and wrist. MyoSuite [2] successfully integrates these into a single MyoHand model, which amalgamates anatomic data from the MoBL, a human upper extremity model [30, 36], and the 2nd-Hand for hand and fingers models [18]. In line with BASH [37] and RCare-World [52], we incorporate the MyoHand model into a parametric MANO model, resulting in our proposed MS-MANO model.

2.2. Visual Hand Dynamics Analysis

Analyzing hand dynamics visually typically involves estimating hand pose from a single image. The considerable advancement of learning-based research in this area is largely due to the parametric hand model, MANO [35], and the differentiable layer that enables direct learning of hand parameters and generation of the hand model. Although the extraction of hand pose or kinematic structures from static images has achieved significant success, these methods [1, 28, 47, 48, 50, 51] are inherently incapable of predicting dynamic information. Recently, some studies [6, 8, 10, 12, 15, 32, 42, 45, 53] have begun to investigate the temporal information in videos in order to regularize per-frame prediction. One approach explicitly models temporal information using techniques such as optical flow [12], temporal consistency constraints [25, 45], and graph modeling [4]. Another approach implicitly models temporal information by incorporating learning techniques with recurrent neural networks [15] or transformers [6, 10]. gSDF [6] adopts a signed distance field for both hand and object geometry and extracts the hand model with marching cube algorithm [27]. Then, the extracted hand mesh can be fitted to the MANO model to obtain the joint parameters. Deformer [10] adopts different transformer modules to extract spatial and temporal information. The features for each image are first extracted separately and then fused with a cross-attention, which improves the accuracy of hand pose estimation.

Different from previous works, instead of extracting spatial-temporal information from image observations, we provide a musculoskeletal prior and design a learning framework that uses this prior to adhere to biomechanical constraints.



(a)

(b)

Figure 2. The hill-type muscle. **(a)** Each muscle segment is composed of the contractile element *CE*, the parallel elastic element *PEE*, and the serial elastic element *SEE*. **(b)** Each muscle segment originates from a certain point n_{origin} and ends at $n_{\text{insertion}}$. A joint j connects two bones. Once triggered, the muscle segment can apply torque $\boldsymbol{\tau}$ on the joint.

3. Musculoskeletal MANO, MS-MANO

In this section, we will describe the musculoskeletal MANO (MS-MANO) model. We first give a brief introduction to the muscle model in Sec. 3.1. Then, we describe the muscle adaptation from bone-centric MyoHand data to joint-centric representation in Sec. 3.2.

3.1. Hill-type Muscle Model

Muscles are soft tissues that can generate forces to facilitate joint movements. To model the muscle dynamics behavior, we adopt the Hill-type model [13], which is widely used in biomechanics [39]. In the hill-type model (see Fig. 2), a muscle consists of segments represented by red dashed lines. Each line segment is modeled by three elements: the contractile element *CE*, the parallel elastic element *PEE*, and the serial elastic element *SEE*. Each muscle initiates from a specific point n_{origin} and triggers the muscle fiber, while the insertion points $n_{\text{insertion}}$ act as the remote endpoints and apply torque to the joint.

Thus, the torque for a joint can be calculated by:

$$\boldsymbol{\tau}_{\text{m}} = f(F, x) \left\| (\boldsymbol{q} - \boldsymbol{j}) \times \frac{\boldsymbol{s}_c}{\|\boldsymbol{s}_c\|} \right\|, \tag{1}$$

where F is the contractile force, x is the muscle state, $\boldsymbol{q}$ is the point where the muscle is attached to the bone, $\boldsymbol{j}$ is the joint to apply torque on, and $\boldsymbol{s}_c$ is the muscle segment. The detailed deduction can be referred to [52].

3.2. Joint-centric Muscle Adaptation

In the physical human body, joints consist of either a single bone or a combination of multiple bones. Take the wrist joint, for example; it comprises *the distal ends of the radius and ulna bones, 8 carpal bones, and the proximal segments of the 5 metacarpal bones* (Fig. 3a). The mappings between the joints and the bones are defined by academic consensus on anatomy. In OpenSim, muscle data is documented based on its relative position to the bone it's attached to. To incorporate this muscle data into a joint-centric skinned model like MANO, we need to establish a mapping between muscles and joints. A direct method to achieve this is by using the existing "muscle-to-bone" relationships to formulate the "bone-to-joint" connections.

Muscle-to-Joint Mapping We first analyze the MyoHand model and establish joint name mapping to the MANO model. Then, we transfer the origin point and insertion point of each muscle segment from bone-centric (*i.e.* relative position to a certain bone) to joint-centric (*i.e.* relative position to a certain joint). Specifically, let's denote the set of joints in the MANO model as $\mathcal{M} = \{m_i\}_{i=1}^{n}$, and the set of bone subgroups in the MyoHand model

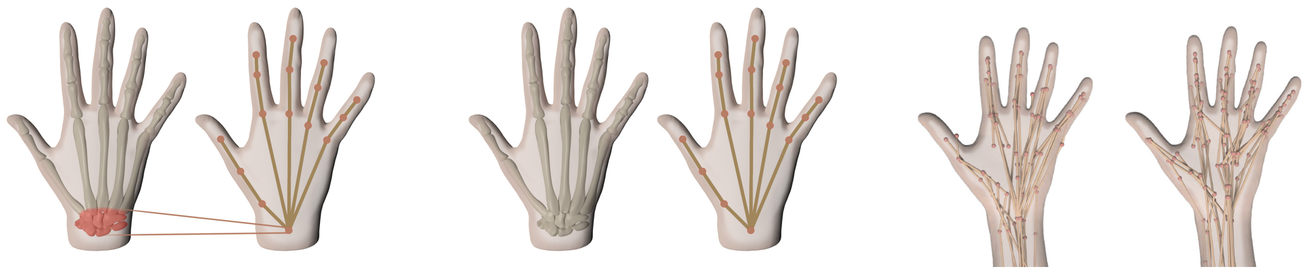


Figure 3. Joint-centric muscle adaptation. **(a)** A set of smaller bones in the MyoHand model is mapped into a single joint in the MANO model. **(b)** The bone-centric muscle segments can adapt to different shapes. **(c)** (Left) The raw skeleton after the automatic mapping will result in issues like intersection. (Right) The manually revised skeleton can perfectly fit with the MANO model.

as $\mathcal{O} = \{O_i = \{b_{i,j}\}_{j=1}^k\}_{i=1}^n$, where n is the number of joints, $\{b_j\}_{j=1}^k$ are the bones in the MyoHand model. This mapping relationship can be represented as a function $f_{\text{mapping}} : O_i \mapsto m_j$.

The location at which the muscle-tendon connects is calculated by taking into account the relative position between the muscle tendon and the geometric mean center of the bone subgroup (which is regarded as the equivalent location of the MANO joint m_j).

For example, consider a point of attachment q which has a displacement relative to a MyoHand bone b_k expressed as $\text{dist}(q, b_k) \in \mathbb{R}^3$, the displacement $\text{dist}(q, m_j)$ relative to a MANO joint m_j after the mapping can be calculated as

$$\boldsymbol{m}_j = \frac{\sum_{o \in O_j} \boldsymbol{o}}{|O_j|}, \tag{2}$$

$$\begin{aligned} \text{dist}(q, m_j) &= \boldsymbol{q} - \boldsymbol{m_j} && (3)\\ &= (\boldsymbol{q} - \boldsymbol{b}_k) + (\boldsymbol{b}_k - \boldsymbol{m}_j) && (4)\\ &= \text{dist}(q, b_k) + \text{dist}(b_k, m_j), && (5) \end{aligned}$$

where $O_j = f^{-1}(m_j)$ represents the subgroup of bones associated with the joint m_j, and the bold symbols represent the location vectors of the bones and points.

Switching from a bone-focused to a joint-focused muscle representation immediately allows the muscle segment to connect solely to the joints. Thus, if the shape changes and alters the joint location, the muscle segment will adjust accordingly, as illustrated in Fig. 3b.

Manual Revision The MyoHand skeleton size may not align perfectly with MANO shapes due to their different human body sources, leading to issues such as muscle tendons intersecting the skin. To address this, we collaborated with human experts to adjust the insertion points slightly. These adjustments ensure anatomical accuracy and compatibility with MyoHand's motion patterns, as shown in Fig. 3c.

Discussion As previously mentioned, the hand's musculoskeletal system is part of the entire upper extremity, with many muscle tendons starting in the forearm and ending in the hand. To incorporate this system into the MANO model, we integrated it with the SMPLX human body model. As shown in Figure 3, our MS-MANO model includes the wrist and forearm. However, for consistency with existing datasets, we only focus on visualizing the hand in later experiments.

4. Biomechanical Pose Refinement Framework, BioPR

For the task of hand pose tracking, we begin with a video sequence $\mathcal{V} = \{I_i\}_{i=1}^t$, which contains a single hand's movements. An off-the-shelf hand pose estimation algorithm is then applied to extract the predicted poses $\mathcal{P}^{\text{pred}} = \{\boldsymbol{p}_i^{\text{pred}}\}_{i=1}^t$ as well as the shape parameters for the MANO model. Subsequently, these predicted poses are interpolated to calculate the velocity of the hand joints at a given time t, represented as $\boldsymbol{v}_t$. By taking the observations near time t, we estimate the excitation signals $\boldsymbol{a}_i$ with an inverse dynamics network, IDNet (Sec. 4.1). We run a forward dynamics simulation with the estimated excitation signals $\boldsymbol{a}$ to get a reference pose $\boldsymbol{p}_i^{\text{ref}}$ and velocity $\boldsymbol{v}_i^{\text{ref}}$. These reference poses and velocities can create valid trajectories, which are beneficial for tasks such as motion generation.

However, for a visual analysis task in this work, due to the natural cross-individual differences in body shape and muscle structure, we cannot rely solely on simulated poses generated from standard muscle parameters. Therefore, we treat the reference pose as a biomechanical constraint and use a small neural network to produce a refined pose in a "simulation-in-the-loop" pipeline (Sec. 4.2).

4.1. Muscle Inverse Dynamics and IDNet

To get the reference pose and velocity within biomechanical constraints, we first need to perform an inverse dynamics process to get the excitation signals $\boldsymbol{a}_i$, followed by a

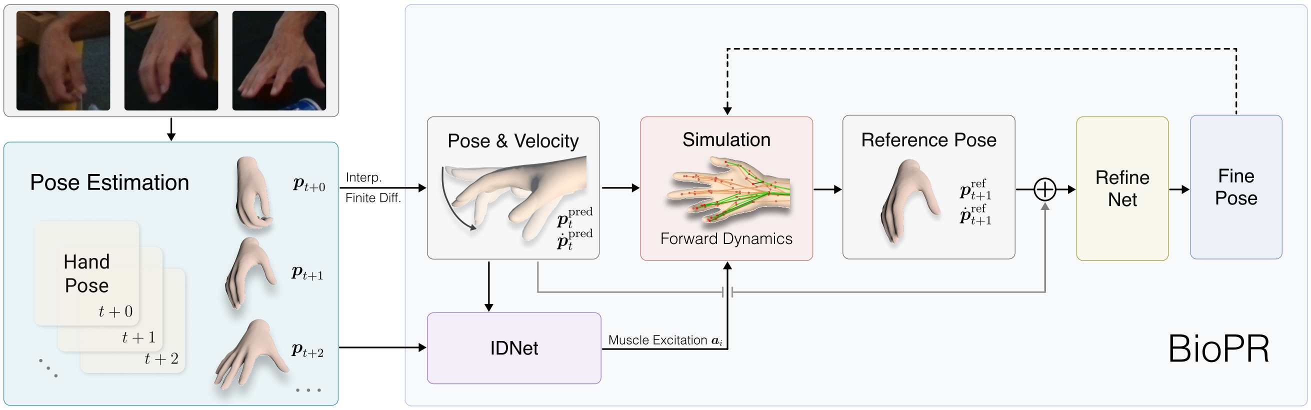


Figure 4. The simulation-in-the-loop pipeline of BioPR. Given a sequence of RGB images and the corresponding predictions of an existing hand pose estimator, BioPR first interpolates and differentiates the poses to get the joint velocities. Then, the IDNet is used to infer the muscle excitation signals. The joint poses, velocities, excitation signals, and the poses of the previous frame (denoted by dotted lines) are sent into the simulator, which will generate the next reference pose by forward kinematics. The Refine Net will do the final refinement based on the pose, velocity, and reference pose. On the next frame, the refined pose can be fed back to the simulator.

forward dynamics process. However, applying the inverse dynamics process to muscle is difficult.

Given a sequence of joint movements represented in axis-angle form as $\boldsymbol{p}_i \in \mathbb{R}^{n_{\text{joint}} \times 3}$ along with its angular velocity $\dot{\boldsymbol{p}}_i = \boldsymbol{v}_i \in \mathbb{R}^{n_{\text{joint}} \times 3}$, as well as the muscle excitation signal $\boldsymbol{a}_i \in \mathbb{R}^{n_{\text{muscle}}}$ that initiated the motion, we define the inverse dynamics modeling for muscle actuators as:

$$f_{\text{inv}}(\boldsymbol{p}_i, \boldsymbol{v}_i, \boldsymbol{p}_{i+1}, \boldsymbol{v}_{i+1}) = \boldsymbol{a}_i, \tag{6}$$

and the corresponding forward dynamics as:

$$f_{\text{fwd}}(\boldsymbol{p}_i, \boldsymbol{v}_i, \boldsymbol{a}_i) = (\boldsymbol{p}_{i+1}, \boldsymbol{v}_{i+1}). \tag{7}$$

In practice, the forward dynamics are computed using a physics engine to accurately simulate the physical interactions and constraints of the system. However, formulating inverse dynamics analytically is challenging, so we use a neural network, **IDNet**, to learn these dynamics.

Owing to the complexity of the human body, it is impossible to obtain accurate ground truth data of muscle excitation signals for each muscle. We instead indirectly supervise these signals by comparing torques. The IDNet produces excitation signals $\boldsymbol{a}_i$, which are then used to compute the torque for each muscle, $\boldsymbol{\tau}_{\text{m}}$, as detailed in Eq. (1). For comparison, the reference torque (treated as ground truth) is calculated using a Proportional-Derivative (PD) controller with inverse dynamics compensation:

$$\boldsymbol{\tau}_{\text{pd}} = k_p(\boldsymbol{p}_d - \boldsymbol{p}) + k_d(\dot{\boldsymbol{p}}_d - \dot{\boldsymbol{p}}), \tag{8}$$

where $\boldsymbol{p}_d, \dot{\boldsymbol{p}}_d$ are the desired pose and velocity.

We adopt a reinforcement learning algorithm, PPO [38], to train the IDNet. The reward function is defined as

$$r = \exp\left(\omega_\tau \cdot \|\boldsymbol{\tau}_{\text{pd}} - \boldsymbol{\tau}_{\text{m}}\|\right),$$

where ω_τ is a constant.

The PD controller is only used to supervise training. In the testing process, the PD controller is disabled. During the testing phase, the PD controller is deactivated, and the muscle actuators are responsible for driving the motions.

4.2. Simulation in the Loop

We have developed an approach called *simulation-in-the-loop* to track hand movements consistently and accurately through a dynamic, iterative process, instead of relying on static estimates. This method simulates the internal dynamics of hand movements by considering the constraints imposed by the musculoskeletal system's biomechanics.

It initiates with the generation of initial hand pose estimates from video sequences using a base pose estimation algorithm. These algorithms are typically MANO-model-based and can produce joint and shape parameters.

Subsequently, we use the finite difference method to compute the derivatives of these estimated poses and result in the angular velocities $\boldsymbol{v}^{\text{pred}} = \dot{\boldsymbol{p}}^{\text{pred}} \in \mathbb{R}^{15 \times 3}$ (the pose and velocity on the wrist joint are ignored). Our pipeline then processes pairs of consecutive pose and velocity data, $(\boldsymbol{p}_i, \boldsymbol{v}_i)$ and $(\boldsymbol{p}_{i+1}, \boldsymbol{v}_{i+1})$, through the IDNet f_{inv} in Eq. (6) to infer the muscle excitation signals $\boldsymbol{a}_i$ that facilitate pose transitions.

At each timestep, the simulator is updated with the current predicted position $\boldsymbol{p}_i^{\text{pred}}$ and velocity $\boldsymbol{v}_i^{\text{pred}}$. The inferred muscle excitation signals are then applied to simulate the next pose. As a result, we obtain a reference pose $\boldsymbol{p}_{i+1}^{\text{ref}}$, which adheres to the constraints of human anatomy and represents what the pose should plausibly be at the next timestep.

In the final stage, both the predicted pose $\boldsymbol{p}_{i+1}^{\text{pred}}$ and the reference pose $\boldsymbol{p}_{i+1}^{\text{ref}}$ are input into a refinement network, a multi-layer perceptron (MLP), which outputs a refined pose estimate, denoted by

$$\boldsymbol{p}^{\text{refined}} = \mathcal{M}(\boldsymbol{p}_{t+1}^{\text{pred}}, \boldsymbol{p}_{t+1}^{\text{ref}}). \tag{9}$$

The loss is defined by comparing with ground truth pose $\boldsymbol{p}_{\text{gt}}$

$$\mathcal{L}_{\text{refine}} = \left\| \boldsymbol{p}^{\text{gt}} - \boldsymbol{p}^{\text{refined}} \right\|. \tag{10}$$

5. Experimental Setting

5.1. Datasets

DexYCB The DexYCB dataset [5] is a large-scale dataset of hand-grasping postures captured using a synchronized setup of 8 cameras. It contains 20 common hand-held objects and 582K annotated 3D hand poses. For our experiments, we use the default `S0` training and testing split provided by the dataset, which separates the data by sequences. The aligned viewpoints and the presence of occlusions of the DexYCB dataset present challenges to evaluating the robustness of our approach.

OakInk The OakInk dataset [49] is a large dataset for understanding hand-object interaction. It has 100 objects and 230k frames of hand poses, captured by a 4-camera setup. The dynamic forces exerted by objects on hands, due to the complex interaction sequences, challenge the stability and accuracy of our interaction simulations. In our experiments, we employ the default `SP0` split of the dataset, which splits the data by camera views.

5.2. Metrics

We use three different metrics to validate the proposed method. The Mean Per Joint Position Error (**MPJPE**) in millimeters is the standard metric for hand pose estimation. It measures the mean joint distance error relative to the hand wrist over all 21 joints. The Area Under the Curve (**AUC**) scores are provided by the official evaluation system of each dataset, which measures the robustness and precision across varying levels of joint thresholds. The Acceleration Error (**AE**) in mm/s^2 is used as a temporal consistency metric following previous works [10, 16].

5.3. Simulation Setup

Our simulation framework is based on the `RFUniverse` platform [9]. The hill-type muscle model is implemented based on the `Kinesis` package in Unity Engine. The human hand is controlled by 31 muscles responsible for facilitating movement beneath the wrist joint, and 8 muscles for controlling the wrist movements. The parameters for maximum isometric force and the length of the contractile element for each muscle are sourced from established MyoHand model data. To accurately capture the interactive dynamics involving the musculoskeletal system, our model incorporates anatomically aligned colliders that conform to the contours of human skeletal structures. Additionally, we introduce a minimal clearance around these colliders to effectively represent the deformable characteristics of human skin during collision events.

5.4. Training Details

IDNet The IDNet is trained using Proximal Policy Optimization (PPO) [38], a common on-policy reinforcement learning method. Its input size is $16 \times 3 \times 4$ and output size is 31, with two 256-d hidden layers. The network is trained on a NVIDIA A40 GPU. We use `RFUniverse` for pose, velocity, and muscle forward dynamics control. The training and simulation pipeline is vectorized. To be specific, we run 128 distributed processes on a platform with 2 AMD EPYC 7763 64-core processors. Each process controls 64 agents. A small Gaussian noise $N(0, 0.1)$ in degree is applied to the joint rotations during the training process.

At each training step, we collect two consecutive frames from the simulator. Therefore, the total batch size is $128 \times 64 \times 2 = 16384$. The learning rate is 3×10^{-4} with an adaptive scheduler [34]. Each process runs at around 100 FPS, so we are able to generate the simulation data at around 10K FPS. It takes approximately 1 hour to train the IDNet.

Refine Net Refine Net employs a Multi-Layer Perceptron (MLP) architecture and is trained sequentially following IDNet's convergence. Its input size is $48 + 45 = 93$ (48 for the $\boldsymbol{p}^{\text{pred}}$ and 45 for the wrist-ignored $\boldsymbol{p}^{\text{refined}}$), and the output size is 48. The network only has a single 64-d hidden layer. We use a learning rate of 1×10^{-3} and a batch size of 10240 for training. We train the network for 4,500 iterations on a single NVIDIA A40 GPU, and it takes about 5 minutes.

6. Results

We evaluate the anatomic accuracy of the MS-MANO model in Sec. 6.1 and the efficacy of BioPR in Sec. 6.2.

6.1. The Anatomic Accuracy of MS-MANO model

To validate the anatomic accuracy of the proposed MS-MANO, we consulted anatomical experts in local hospitals and compared it with MyoHand [2] as it is a comprehensive musculoskeletal hand model built upon genuine anatomic data [18, 30, 36]. Figure Fig. 6 presents the time-position plots for the ring finger bending trajectory in the MyoHand and MS-MANO models, monitoring three joints. The graphs indicate similar movement patterns in both models, with our model exhibiting a quicker response to muscle stimulation. This implies that our model activates faster, mirroring the immediate response of human muscles to neural signals, even at the extremes of joint motion.

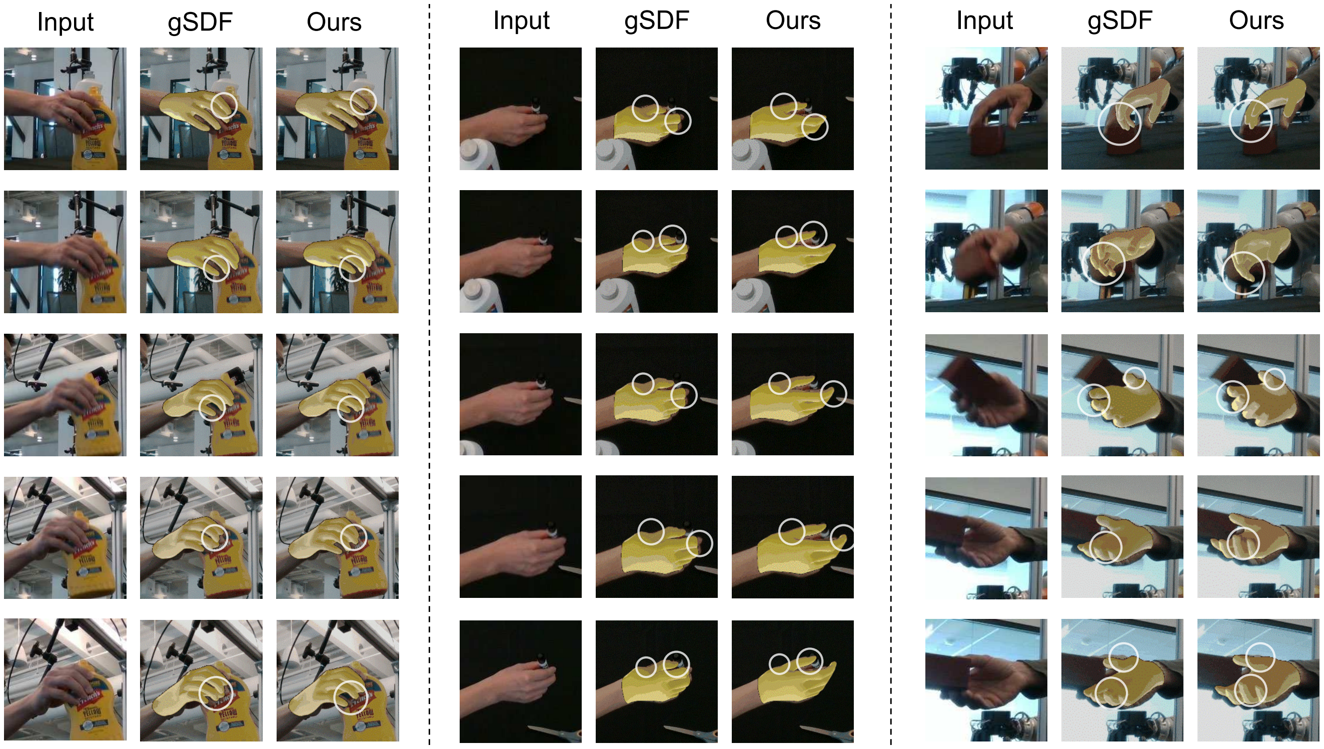
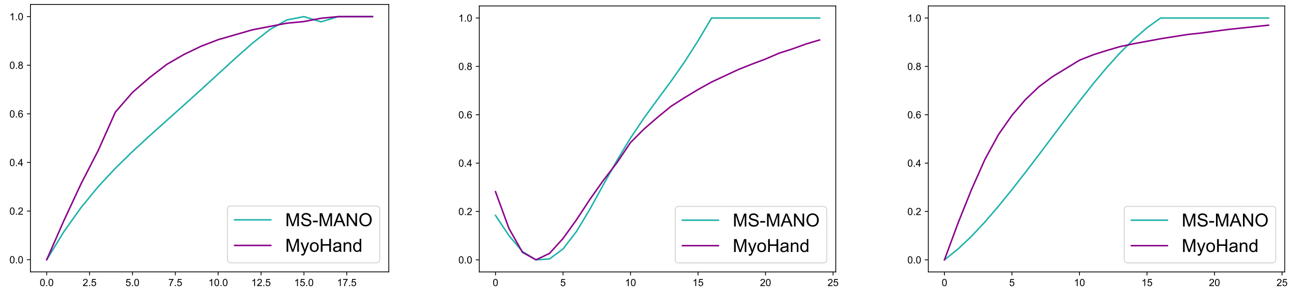


Figure 5. Qualitative results on DexYCB. **Left**: When a person is forcefully grasping a mustard bottle, there is a difference in the tightness of the middle, ring, and little fingers, comparing gSDF to our method. The projected results of our method better align with the input image. **Middle**: The thumb posture predicted by the gSDF method exhibits some odd distortion, which is not observed in our approach. **Right**: When there is severe occlusion, gSDF may generate some hand poses that lead to punctuation with the object. Our method mitigates such problem by catching the dynamics of the musculoskeletal system.



(a) Joint Ring 1 Movements

(b) Joint Ring 2 Movements

(c) Joint Ring 3 Movements

Figure 6. The accuracy of simulation. The figures show normalized joint movements when exciting the FDS4_R muscle, which can drive the ring finger to bend. The x axis is the frames, and the y axis is the relative joint movement.

6.2. Comparison with Baselines on DexYCB

We compare the proposed method with some state-of-the-art methods. Quantitative results on DexYCB are reported in Tab. 1. It shows that our method outperforms previous baseline methods. BioPR consistently and effectively enhances the base hand pose estimation method by refining the prediction results with the biomechanical constraints. Notably, the 3-layer IDNet and the 2-layer RefineNet have only 0.1M and 0.01M parameters, respectively. The BioPR framework only adds a minimal computational load to the base models, with an inference cost of just 9 ms.

Qualitative results are illustrated in Fig. 5, which shows the stability of the bio correction. We visualize our results by projecting the predicted 3D meshes onto the input images. Our approach demonstrates enhanced robustness in predicting poses that are aligned with anatomical structures. In contrast, previous methods exhibit certain artifacts, particularly unnatural twisting of thumb joints.

Methods	MPJPE↓	AUC↑	AE↓
VIBE [16]	16.95	67.5	36.4
TCMR [7]	16.03	70.1	34.3
MeshGraphormer [24]	16.21	69.1	35.9
gSDF [6]	14.4	89.1	30.3
gSDF + BioPR	**12.81**	**89.7**	**29.9**
Deformer [10]	13.64	89.6	31.7
Deformer + BioPR	**12.92**	**90.4**	**30.7**

Table 1. Quantitative results on DexYCB.

6.3. Failure Cases

Incorrect Annotations Some of the failures come from the incorrect annotations of the ground truth. Since the annotations do not properly match the visual representation of the hand in the image, training our method on the incorrect ground truth data leads to a misaligned result.

Errors from the Base Models Occlusion, lighting, and various other factors can cause the base models to inaccurately predict results from the provided images. These inaccuracies can be persistent, enduring throughout the whole sequence. Our BioPR has the ability to correct it to some degree, but it still depends on the initial predictions from the base models to produce valid results.

Extremely Slow Motion A significant number of hand motion sequences exhibit minimal temporal variations. In these cases, the muscles are only slightly stimulated and generate small torques. These minor excitation signals fail to produce sufficient movement to achieve accurate frame-to-frame offsets, making the dynamics estimation and analysis challenging.

6.4. Ablation Study

Heuristics-based Priors To improve pose estimation in videos, where analyzing every frame is often unnecessary, we sample frames at intervals. This approach might lead to inconsistent motion predictions. We explored temporal smoothing, PCA, and TOCH [53] to mitigate this. Temporal smoothing averages poses within a window size d. The smoothed pose $\hat{\boldsymbol{p}}_i$ for a frame pose $\boldsymbol{p}_i \in I$ is determined by:

$$\hat{\boldsymbol{p}}_i = \arg\min_{\boldsymbol{m}} \sum_{j=i-d/2}^{i+d/2} \|\boldsymbol{p}_j - \boldsymbol{m}\|_2^2. \tag{11}$$

Temporal smoothing and TOCH do not outperform our method, showing less than a 0.8mm improvement in MPJPE using gSDF with DexYCB dataset. PCA, on the other hand, led to a 7.91mm increase in MPJPE. These manually designed methods may not fully capture the complex dynamics due to the high dexterity of human hands.

However, our research shows that smoothing, while not significantly improving pose accuracy, creates realistic movement speeds. This outcome reduces sudden changes in muscle stimulation, resulting in smoother motion paths.

Muscle Insertion Points The muscles are highly sensitive to their insertion points [29]. Even minor alterations in these attachment points can lead to significant shifts in the trajectory of joint movement. We have experimented with various configurations of these insertion points and compared the trajectories before and after manual revision. The detailed visualization is presented in the supplementary materials. Such variations can cause the **IDNet** to struggle with convergence, at the same time resulting in noticeable artifacts within the joint movements.

7. Conclusion

In conclusion, our study presents an innovative approach to enhance hand dynamics analysis by integrating the musculoskeletal system into the learnable parametric hand model, MANO, resulting in the MS-MANO model. This model allows for movements that are more human-like and physiologically realistic and bridges the gap between image observations to biomechanics. The BioPR refiner further refines hand pose estimations. Despite the challenges in creating the musculoskeletal model introduced by the complexity of the hand's dynamics system, our work validates the muscle adaptation by comparing the generated joint torques with MyoSuite. Moreover, the MS-MANO model and BioPR's effectiveness are evaluated on two large-scale public datasets, DexYCB and OakInk. The results showed consistent improvement in both quantitative and qualitative measures. Therefore, MS-MANO and BioPR mark a significant advancement in visual hand dynamics analysis, opening new avenues for future research and applications.

Acknowledgements

We thank Yanbing Zhou for coding and figure support, Chen Bao for designing the initial codebase, Danqing Li and Yongkai Fan for muscle data processing, Yuan Fang and Haoyuan Fu for experimental help, and Jieyi Zhang, Wenxin Du and Han Xue for their insightful discussions.

This work was supported by the National Key Research and Development Project of China (No. 2022ZD0160102), National Key Research and Development Project of China (No. 2021ZD0110704), Shanghai Artificial Intelligence Laboratory, XPLORER PRIZE grants.

References

[1] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1067–1076, 2019. 1, 3

[2] Vittorio Caggiano, Huawei Wang, Guillaume Durandau, Massimo Sartori, and Vikash Kumar. Myosuite–a contact-rich simulation suite for musculoskeletal motor control. *arXiv preprint arXiv:2205.13600*, 2022. 2, 6

[3] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2

[4] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2272–2281, 2019. 3

[5] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9044–9053, 2021. 2, 6

[6] Zerui Chen, Shizhe Chen, Cordelia Schmid, and Ivan Laptev. gsdf: Geometry-driven signed distance functions for 3d hand-object reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12890–12900, 2023. 1, 2, 3, 8

[7] Hongsuk Choi, Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Beyond static features for temporally consistent 3d human pose and shape from a video. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 8

[8] Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J Black, and Otmar Hilliges. Arctic: A dataset for dexterous bimanual hand-object manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12943–12954, 2023. 3

[9] Haoyuan Fu, Wenqiang Xu, Ruolin Ye, Han Xue, Zhenjun Yu, Tutian Tang, Yutong Li, Wenxin Du, Jieyi Zhang, and Cewu Lu. Demonstrating rfuniverse: A multiphysics simulation platform for embodied ai. 6

[10] Qichen Fu, Xingyu Liu, Ran Xu, Juan Carlos Niebles, and Kris M Kitani. Deformer: Dynamic fusion transformer for robust hand pose estimation. *arXiv preprint arXiv:2303.04991*, 2023. 1, 2, 3, 6, 8

[11] Thomas Geijtenbeek, Michiel Van De Panne, and A Frank Van Der Stappen. Flexible muscle-based locomotion for bipedal creatures. *ACM Transactions on Graphics (TOG)*, 32(6):1–11, 2013. 2

[12] Yana Hasson, Bugra Tekin, Federica Bogo, Ivan Laptev, Marc Pollefeys, and Cordelia Schmid. Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 571–580, 2020. 3

[13] Archibald Vivian Hill. The heat of shortening and the dynamic constants of muscle. *Proceedings of the Royal Society of London. Series B-Biological Sciences*, 1938. 3

[14] Yifeng Jiang, Tom Van Wouwe, Friedl De Groote, and C Karen Liu. Synthesis of biologically realistic human motion using joint torque actuation. *ACM Transactions On Graphics (TOG)*, 38(4):1–12, 2019. 1

[15] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5253–5263, 2020. 3

[16] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. Vibe: Video inference for human body pose and shape estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6, 8

[17] Taku Komura, Yoshihisa Shinagawa, and Tosiyasu L Kunii. Creating and retargetting motion by the musculoskeletal human body model. *The visual computer*, 16:254–270, 2000. 2

[18] Jong Hwa Lee, Deanna S Asakawa, Jack T Dennerlein, and Devin L Jindrich. Finger muscle attachments for an opensim upper-extremity model. *PloS one*, 10(4):e0121712, 2015. 2, 6

[19] Seunghwan Lee, Ri Yu, Jungnam Park, Mridul Aanjaneya, Eftychios Sifakis, and Jehee Lee. Dexterous manipulation and control with volumetric muscles. *ACM Transactions on Graphics (TOG)*, 37(4):1–13, 2018. 2

[20] Seunghwan Lee, Moonseok Park, Kyoungmin Lee, and Jehee Lee. Scalable muscle-actuated human simulation and control. *ACM Transactions On Graphics (TOG)*, 38(4):1–13, 2019. 2

[21] Sung-Hee Lee and Demetri Terzopoulos. Heads up! biomechanical modeling and neuromuscular control of the neck. In *ACM SIGGRAPH 2006 Papers*, pages 1188–1198. 2006. 2

[22] Sung-Hee Lee, Eftychios Sifakis, and Demetri Terzopoulos. Comprehensive biomechanical modeling and simulation of the upper body. *ACM Transactions on Graphics (TOG)*, 28 (4):1–17, 2009. 2

[23] Yoonsang Lee, Moon Seok Park, Taesoo Kwon, and Jehee Lee. Locomotion control for many-muscle humanoids. *ACM Transactions on Graphics (TOG)*, 33(6):1–11, 2014. 2

[24] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *ICCV*, 2021. 8

[25] 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

[26] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM TOG*, 34(6):248:1–248:16, 2015. 2

[27] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM TOG*, 21(4):163–169, 1987. 3

[28] Jun Lv, Wenqiang Xu, Lixin Yang, Sucheng Qian, Chongzhao Mao, and Cewu Lu. Handtailor: Towards high-precision monocular 3d hand recovery. 2021. 1, 3

[29] Shihan Ma, Irene Mendez Guerra, Arnault Hubert Caillet, Jiamin Zhao, Alexander Kenneth Clarke, Kostiantyn Maksymenko, Samuel Deslauriers-Gauthier, Xinjun Sheng, Xiangyang Zhu, and Dario Farina. Neuromotion: Open-source simulator with neuromechanical and deep network models to generate surface emg signals during voluntary movement. 2023. 8

[30] Daniel C McFarland, Emily M McCain, Michael N Poppo, and Katherine R Saul. Spatial dependency of glenohumeral joint stability during dynamic unimanual and bimanual pushing and pulling. *Journal of biomechanical engineering*, 141 (5):051006, 2019. 2, 6

[31] Igor Mordatch, Jack M Wang, Emanuel Todorov, and Vladlen Koltun. Animating human lower limbs using contact-invariant optimization. *ACM Transactions on Graphics (TOG)*, 32(6):1–8, 2013. 2

[32] JoonKyu Park, Yeonguk Oh, Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. Handoccnet: Occlusion-robust 3d hand mesh estimation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1496–1505, 2022. 3

[33] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, 2019. 2

[34] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. 6

[35] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 2017. 1, 2

[36] Katherine R Saul, Xiao Hu, Craig M Goehler, Meghan E Vidt, Melissa Daly, Anca Velisar, and Wendy M Murray. Benchmarking of dynamic simulation predictions in two software platforms using an upper limb musculoskeletal model. *Computer methods in biomechanics and biomedical engineering*, 18(13):1445–1458, 2015. 2, 6

[37] Robert Schleicher, Marlies Nitschke, Jana Martschinke, Marc Stamminger, Björn M Eskofier, Jochen Klucken, and Anne D Koelewijn. Bash: Biomechanical animated skinned human for visualization of kinematics and muscle activity. In *VISIGRAPP (1: GRAPP)*, pages 25–36, 2021. 2

[38] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. 5, 6

[39] Ajay Seth, Jennifer L Hicks, Thomas K Uchida, Ayman Habib, Christopher L Dembia, James J Dunne, Carmichael F Ong, Matthew S DeMers, Apoorva Rajagopal, Matthew Millard, et al. Opensim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. *PLoS computational biology*, 14(7):e1006223, 2018. 2, 3

[40] Weiguang Si, Sung-Hee Lee, Eftychios Sifakis, and Demetri Terzopoulos. Realistic biomechanical simulation and control of human swimming. *ACM Transactions on Graphics (TOG)*, 34(1):1–15, 2014. 2

[41] Shinjiro Sueda, Andrew Kaufman, and Dinesh K Pai. Musculotendon simulation for hand animation. In *ACM SIGGRAPH 2008 papers*, pages 1–8. 2008. 2

[42] Garvita Tiwari, Dimitrije Antić, Jan Eric Lenssen, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. Pose-ndf: Modeling human pose manifolds with neural distance fields. In *European Conference on Computer Vision*, pages 572–589. Springer, 2022. 3

[43] Winnie Tsang, Karan Singh, and Eugene Fiume. Helping hand: an anatomically accurate inverse dynamics solution for unconstrained hand motion. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 319–328, 2005. 2

[44] Jack M Wang, Samuel R Hamner, Scott L Delp, and Vladlen Koltun. Optimizing locomotion controllers using biologically-based actuators and objectives. *ACM Transactions on Graphics (TOG)*, 31(4):1–11, 2012. 2

[45] John Yang, Hyung Jin Chang, Seungeui Lee, and Nojun Kwak. Seqhand: Rgb-sequence-based 3d hand pose and shape estimation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, pages 122–139. Springer, 2020. 3

[46] Lixin Yang, Jiasen Li, Wenqiang Xu, Yiqun Diao, and Cewu Lu. Bihand: Recovering hand mesh with multi-stage bisected hourglass networks. In *BMVC British Machine Vision Conference*, 2020. 1

[47] Lixin Yang, Xinyu Zhan, Kailin Li, Wenqiang Xu, Jiefeng Li, and Cewu Lu. Cpf: Learning a contact potential field to model the hand-object interaction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11097–11106, 2021. 3

[48] Lixin Yang, Kailin Li, Xinyu Zhan, Jun Lv, Wenqiang Xu, Jiefeng Li, and Cewu Lu. Artiboost: Boosting articulated 3d hand-object pose estimation via online exploration and synthesis. In *CVPR IEEE Conference on Computer Vision and Pattern Recognition*, pages 2750–2760, 2022. 3

[49] Lixin Yang, Kailin Li, Xinyu Zhan, Fei Wu, Anran Xu, Liu Liu, and Cewu Lu. Oakink: A large-scale knowledge repository for understanding hand-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20953–20962, 2022. 2, 6

[50] Lixin Yang, Xinyu Zhan, Kailin Li, Wenqiang Xu, Junming Zhang, Jiefeng Li, and Cewu Lu. Learning a contact potential field for modeling the hand-object interaction. *IEEE transactions on pattern analysis and machine intelligence*, 2024. 3

[51] Ruolin Ye, Wenqiang Xu, Zhendong Xue, Tutian Tang, Yanfeng Wang, and Cewu Lu. H2o: A benchmark for visual human-human object handover analysis. In *ICCV IEEE/CVF International Conference on Computer Vision*, pages 15762–15771, 2021. 3

[52] Ruolin Ye, Wenqiang Xu, Haoyuan Fu, Rajat Kumar Jenamani, Vy Nguyen, Cewu Lu, Katherine Dimitropoulou, and Tapomayukh Bhattacharjee. Rcare world: A human-centric simulation world for caregiving robots. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 33–40. IEEE, 2022. 2, 3

[53] Keyang Zhou, Bharat Lal Bhatnagar, Jan Eric Lenssen, and Gerard Pons-Moll. Toch: Spatio-temporal object-to-hand correspondence for motion refinement. In *European Conference on Computer Vision*, pages 1–19. Springer, 2022. 3, 8