2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

HandyPriors: Physically Consistent Perception of Hand-Object Interactions with Differentiable Priors

Shutong Zhang[1,*], Yi-Ling Qiao[2,*], Guanglei Zhu[1,*], Eric Heiden[3], Dylan Turpin[1,3], Jingzhou Liu[1], Ming Lin[2,4], Miles Macklin[3], Animesh Garg[1,3]

***Abstract*— Various heuristic objectives for modeling hand-object interaction have been proposed in past work. However, due to the lack of a cohesive framework, these objectives often possess a narrow scope of applicability and are limited by their efficiency or accuracy. In this paper, we propose HANDYPRIORS, a unified and general pipeline for pose estimation in human-object interaction scenes by leveraging recent advances in differentiable physics and rendering. Our approach employs rendering priors to align with input images and segmentation masks along with physics priors to mitigate penetration and relative-sliding across frames. Furthermore, we present two alternatives for hand and object pose estimation. The optimization-based pose estimation achieves higher accuracy, while the filtering-based tracking, which utilizes the differentiable priors as dynamics and observation models, executes faster. We demonstrate that HANDYPRIORS attains comparable or superior results in the pose estimation task, and that the differentiable physics module can predict contact information for pose refinement. We also show that our approach generalizes to perception tasks, including robotic hand manipulation and human-object pose estimation in the wild.**

I. INTRODUCTION

Hands serve as the primary means by which humans interact with the physical world as they are responsible for most of the intricate and dexterous object manipulation tasks. A good understanding of hand-object interaction helps with numerous learning-based tasks such as action recognition [1], robotic manipulation [2], user interaction [3], etc.

Estimating hand and object interaction presents several challenges. Firstly, the complex, deformable geometry and dynamics of hands present challenges in learning and optimization. Unlike rigid objects, hands are articulated with joints and possess high degrees of freedom. Their non-linearity and non-convexity further exacerbate the complexity of the task. Secondly, the presence of both self-occlusion and hand-object occlusion result in uncertainty when predicting hand poses during manipulation. Finally, the rich contacts between hands and objects in interaction scenes necessitate the need for a consistent estimation method for both the hand and the object to avoid penetration.

There are generally two categories of approaches for estimating hand and object poses. The first approach is purely learning-based, where the network architectures, loss functions, and training strategies are carefully designed to train a generalizable mapping from images to hand and object configurations [4], [5]. Although neural networks and large training datasets can mitigate the difficulties posed by high degrees of freedom and non-convexity, they cannot guarantee interpretability nor correctness. The second category of approaches is optimization-based [6], [7], which involves defining and optimizing heuristic target functions to improve physical feasibility and reduce penetrations. Although the optimization process may be slower, per-frame iterations can improve the accuracy of the results.

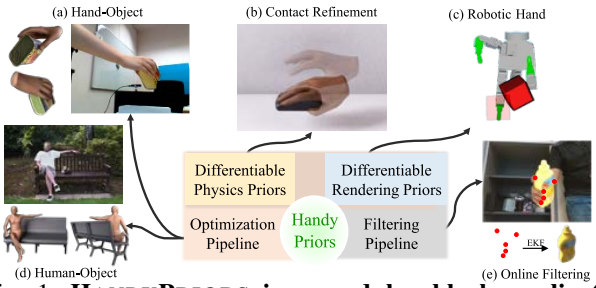


Fig. 1: **HANDYPRIORS is a modular block application in multiple tasks.** The estimation of hand-object interaction can be achieved by (a) optimization-based refinement or (e) filtering with Extended Kalman Filter (EKF). Moreover, the differentiable contact module can be used to perform (b) pose refinement given contact status. Our differentiable priors can also be used in pose estimation for scenes with (c) robot hands or (d) human bodies.

In this paper, we leverage both learning and optimization techniques. Our method utilizes pre-trained networks to provide an initial estimate of poses, and then uses differentiable self-supervised terms to further refine the estimation by correcting errors from pixel-level misalignment and non-physical penetration/oscillation. Inspired by the success of self-supervised learning in large-scale vision and language models [8], [9], we adopt similar techniques to estimate hand-object interactions. Specifically, we propose *supervising the pose prediction by comparing it with subsequent video frames*, similar to the way in which text generation can be trained using sequence completion.

To enable backpropagation of gradients from image space, we develop *fully-differentiable operators that simulate and render predicted poses*. This closed-loop optimization approach improves the accuracy by making the estimation resemble more of the input image. We also *incorporate several regularization terms to minimize motion oscillation and collision in predicted sequences*.

While the optimization process improves the accuracy, it can be time-consuming. To address this issue, we present an

[1]University of Toronto & Vector Institute, [2]University of Maryland College Park, [3]Nvidia, [4]Amazon. *Equal contribution

More experiments and supplementary materials can be found at: https://handypriors.github.io/

alternative filtering technique based on the Kalman filter [10] to further *utilize our differentiable dynamics and observation modules*. This approach allows the user to make a trade-off between efficiency and accuracy depending on whether they choose to perform online tracking or offline optimization. Employing this integrated differentiable rendering and simulation pipeline, we also apply our pose estimation method to human and synthetic robot hand scenes, demonstrating the generalizability and robustness of our method. Figure 1 summarizes several applications of this approach.

In summary, the key contributions of this work are:

- An **integrated differentiable rendering** (Sec. III-B) and **simulation** (Sec. III-C) pipeline to estimate hand-object interaction.
- An **offline optimization** (Sec. III-D) and an **online tracking** (Sec. III-E) process that can predict poses and contact information, providing the user with options to balance between accuracy and efficiency.
- **Experiments that generalize our priors to applications**, e.g. robotic manipulation (Sec. IV-A), contact refinement (Sec. IV-D), and human bodies (Sec. IV-E).

In our experiments, our method achieves comparable or better pose estimation results in hand-object-interaction datasets [11], [12]. Additionally, we show that the contact information predicted by our method effectively refines pose estimation. Our ablation study compares the differences between the optimization and tracking results, further validating the contributions of different losses.

II. Related Works

Hand and object pose estimation from monocular images [13], [14], [15], [16] is a fundamental problem in computer vision. Many recent works predict 6 DoF object poses assuming known rigid mesh models [17], [18], [19]. For human hands, MANO [20] proposes a parametric model to represent the hand geometry using different shape and pose parameters. Numerous learning-based approaches focus on predicting 3D joint locations from RGB(D) inputs [21], [22], [23], [24], [25], [26], [27]. Others [28], [29], [7], [30] then regress to the MANO pose and shape parameters. There are also approaches directly learning a mapping between visual inputs and hand poses from annotated datasets [31], [32], [33], [34]. These methods achieved remarkable success on hand-only datasets but had poor performance when hands were holding objects. There are also optimization-based approaches that optimize hand parameters based on 2D key points [6], [7], [35] or segmentations [11]. They tend to have better accuracy but are much slower due to the iterative optimization process. We propose a tracking-based pipeline with higher efficiency. Our differentiable priors are *more general* and can be applied to a *wider range* of applications.

Hand object interaction has been garnering progressively more attention. Treating hand and object pose estimation separately results in inaccurate estimation for both. With an increasing number of hand-object datasets being created [36], [37], [38], [11], [4], [39], many recent works have started to focus on jointly tracking hands and objects. However, simultaneously estimating both is challenging due to occlusions and depth ambiguity. [5] designed a joint learning framework based on contextual reasoning between hand and object representations. [40] proposed an ordinal relation loss to correct depth misalignment between hand and object. [41] incorporated a feature injection mechanism that better handles occlusion by integrating hand information into the obscured areas. These methods improve the accuracy of pose estimation but pay little attention to the physical interaction between hand and objects. Other works aim to predict physically plausible hand-object poses. To achieve this goal, interaction constraints are applied to encourage contact and minimize interpenetration. [7], [6] proposed to use signed distance functions (SDF) to model contact between hand and object, [42] design a deep network to estimate contact areas and a virtual capsule technique to simulate soft hand tissue deformation. [43] apply motion and force vectors to reconstruct interactions. Compared to these methods, our approach is more general and can be used in *contact-based refinement and pose estimation in both human-hand and non-human-hand scenarios*.

Differentiable Priors. Differentiable rendering [44], [45], [46] and physics [47], [48], [49] have been used to solve inverse problems for their efficiency in optimization. For hand grasping, [50] leverages a physics simulator to evaluate dynamic interactions between the hand and the object. [51] uses differentiable physics to generate physically realistic and stable grasps. There are also recent works that use differentiable rendering to optimize human [52] or object [53] poses. However, they do not consider their physical interactions. [54] utilize a differentiable articulated body simulator to learn and smoothen human-pose estimation. In contrast, our method leverages rendering priors for each frame, thereby significantly accelerating pose estimation in a video.

III. Methods: HandyPriors

A. Problem Setting

Given a sequence of RGB hand-object-interaction images and the object's 3D model, we propose two pipelines to estimate the configuration of the hand and object in each frame. Figure 2 shows an overview of our method.

In particular, we represent the hand using the MANO [20] parameterization, which describes a hand by its pose $\theta \in \mathbb{R}^{45+6}$ (representing the rotations of 15 hand joints and a 6 DoF global transformation) and shape $\beta \in \mathbb{R}^{10}$ (a statistical model) parameters. The 778 vertices $\mathbf{V}_{hand} \in \mathbb{R}^{778\times 3}$ of the hand mesh can be computed from θ, β using a learned linear model $\mathbf{V}_{hand} = m(\theta, \beta)$.

The object pose has 6 degrees of freedom, which contains its translation $\mathbf{t} \in R^3$ and rotation (represented by XYZ Euler angles) $\mathbf{r} \in R^3$.

B. Differentiable Rendering Priors

The first prior of our approach is differentiable rendering, which helps align the estimated 3D model with the 2D image. Instance segmentation is the key information we rely on

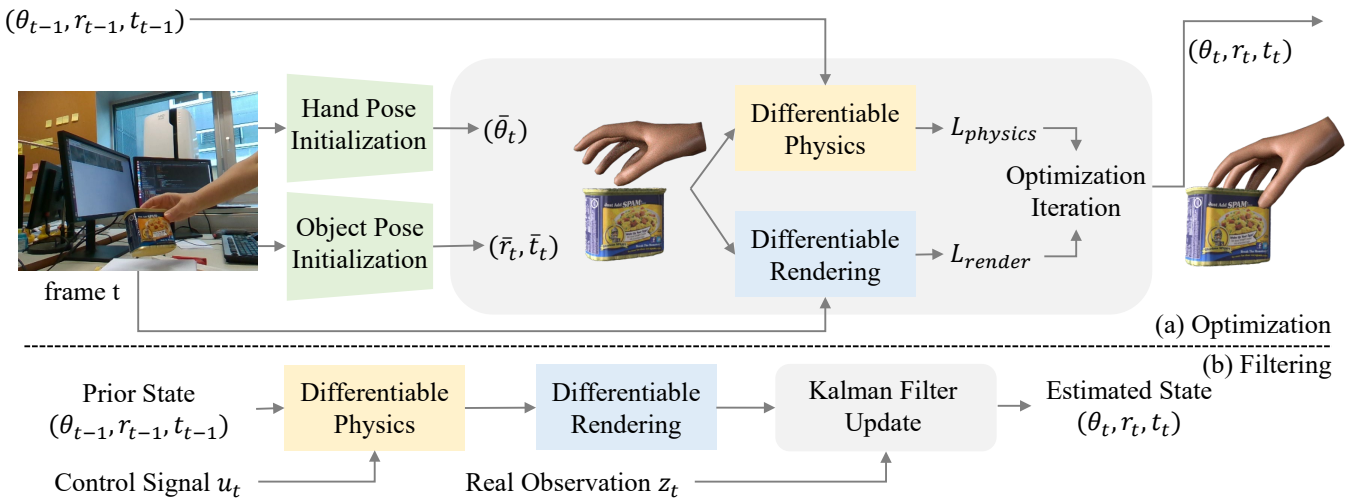


Fig. 2: **An overview of our optimization and filtering pipelines.** We provide two alternatives for utilizing the differentiable priors. Given the image and the estimation from previous frames, (a) the optimization pipeline first initializes the poses with pre-trained networks and then minimizes the rendering and physics losses from the differentiable operators; (b) the filtering pipeline can take some simple observations and use Extended Kalman Filter (EKF) to update the state estimation. EKF requires differentiable physics and rendering to model the system and runs much faster than the optimization pipeline.

to optimize hand and object poses. We use a segmentation detector [55] trained on the COCO dataset [56] to extract hand and object masks. Our experiment shows that although the image segmentation detector, *PointRend* [55], was not trained on the YCB objects [57], it can correctly predict object masks. However, it fails to adequately segment hands as fingertips are harder to detect. Thus, we further trained *PointRend* to obtain better segmentations for the hand and hand-held objects.

For each frame, we utilize a differentiable renderer *SoftRas* [44] that rasterizes and projects 3D hand and object models to 2D, then we optimize hand parameters θ and object parameters $\mathbf{r}, \mathbf{t}$ by fitting the projected vertices onto the hand and object masks. Next, we describe loss terms related to the rendering part.

RGB image loss. Since hand texture varies greatly from person to person, we only render the RGB images for objects. We then apply L2 loss between the rendered image I_r and the input RGB image I_{in} cropped by the detected object mask M_{obj}. The image loss term can be expressed as follow:

$$L_{img}(\mathbf{r}, \mathbf{t}) = \|(1 - M_{hand})I_r - M_{obj}I_{in}\|_2^2 \quad (1)$$

Mask loss. Compared to the RGB image loss, masks have fewer features but are not sensitive to lighting and texture, hence the loss based on masks is more robust and general. We use a differentiable renderer to render the silhouette of hands and objects together. We then apply L2 loss between the rendered mask M_r and the detected mask M_d,

$$L_{mask}(\theta, \mathbf{r}, \mathbf{t}) = \|M_r - M_d\|_2^2 \quad (2)$$

C. Differentiable Physics Priors

Besides the rendering priors defined on the 2D image spaces, we also add more physics-related priors in the 3D and temporal domain to help us regularize the hand-object understanding.

Contact loss. During manipulation, we observe that most of the contacting vertices between hand and objects do not have sudden relative sliding. Based on this observation, we first utilize a GPU-based differentiable contact module to infer the signed distance function $sdf(\mathbf{V}_{hand})$ from hand vertices to the object. Then, we find the contact points $\mathbf{v}_t$ where the SDF values are smaller than 0.1. The relative sliding between t and $t-1$ is,

$$L_{sliding}(\theta, \mathbf{r}, \mathbf{t}) = \|(\mathbf{v}_t - \mathbf{t}_t)(\mathbf{r}_t^{-1})^\top - (\mathbf{v}_{t-1} - \mathbf{t}_{t-1})(\mathbf{r}_{t-1}^{-1})^\top\|$$

Moreover, given the SDF results, we reduce penetration by penalizing the negative SDF values.

$$L_{penetration}(\theta, \mathbf{r}, \mathbf{t}) = \min(0, -sdf(\mathbf{V}_{hand})) \quad (3)$$

Smoothness loss. To enhance the continuity of hand and object across time, we penalize sudden changes of object and hand in two consecutive frames. For objects, we apply L2 loss between object vertices in the current frame $\mathbf{V}_t$ and previous frame $\mathbf{V}_{t-1}$. For hands, we compute the L2 loss between hand poses in two frames θ_t, θ_{t-1},

$$L_{con} = \|\mathbf{V}_t - \mathbf{V}_{t-1}\|_2^2 + \|\theta_t - \theta_{t-1}\|_2^2. \quad (4)$$

During the experiment, we also observed that objects usually have jittering for several frames. Thus, we add a smoothness term to punish the oscillation in object velocity,

$$L_{smo} = \|(\mathbf{V}_t - \mathbf{V}_{t-1}) - (\mathbf{V}_{t-1} - \mathbf{V}_{t-2})\|_2^2. \quad (5)$$

D. Optimization-based Refinement

In summary, the losses of rendering priors are

$$L_{render} = \lambda_1 L_{image} + \lambda_2 L_{mask}, \quad (6)$$

and the physics-inspired losses are

$$L_{physics} = \lambda_3 L_{sliding} + \lambda_4 L_{penetration} + \lambda_5 L_{con} + \lambda_6 L_{smo},$$

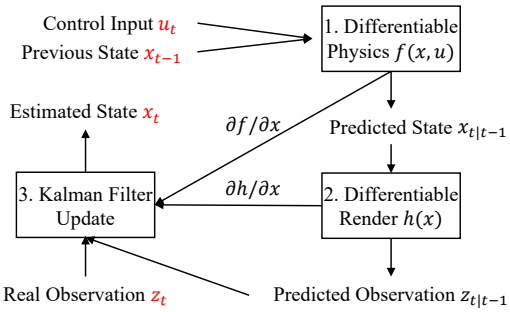


Fig. 3: **Diagram of the EKF filtering process.** With (1) the differentiable physics f and (2) rendering h modules to compute the Jacobians, (3) the extended Kalman Filter (EKF) can perform state estimation given the control signal, observation, and previously estimated states.

TABLE I: **Ablation study.** We run ablation studies for optimization (left) and filtering (right) pipelines. For optimization, 2D and 3D errors (cm) decrease with gradually added rendering $\mathcal{L}_{render}$ and physics loss $\mathcal{L}_{physics}$. Filtering (0.03 s) is faster than the optimization (e.g. ours take 5 s and Homan [7] takes 9 s) but sacrifices accuracy.

Optimization	2D Error	3D Error	Runtime (s)	Filtering	2D Error	3D Error	Runtime (s)
Initialization	1.03	52.5	0.072	$\mathcal{O}_{hand}$	44.59	**5.57**	**0.024**
$+\mathcal{L}_{render}$	0.77	1.54	2.29	$\mathcal{O}_{object}$	19.00	11.24	0.027
$+\mathcal{L}_{physics}$	**0.43**	**1.10**	5.83	$\mathcal{O}_{hand+object}$	**8.84**	7.41	0.029

where λ's are the weights for the loss terms. With these differentiable loss terms, we optimize the pose of the hand θ and object $\mathbf{r}, \mathbf{t}$ using a gradient-based optimization method such as Adam [58].

E. Filtering-based Tracking

Besides the optimization-based process, we can also use the differentiable priors in a more light-weight (faster but less accurate) pipeline (Fig. 3). Multiple filtering algorithms (e.g. Extended Kalman Filter (EKF) [10]) can be integrated with this method for fast tracking. EKF can give a state estimation in a dynamic system

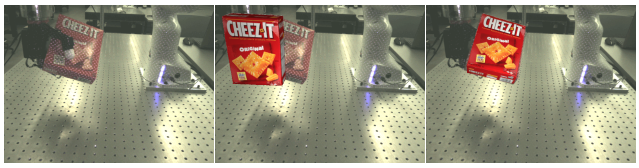
$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t) + \epsilon_s, \mathbf{z}_t = h(\mathbf{x}_t) + \epsilon_o, \tag{7}$$

where $\mathbf{x}$ is the state, $\mathbf{u}$ is the control, $\mathbf{z}$ is the observation, ϵ_s and ϵ_o are noises, $f(\cdot), h(\cdot)$ are *differentiable* functions for states and observation, respectively. Using the EKF, we can compute an estimation $\hat{\mathbf{x}}_t$ for the hidden state $\mathbf{x}_t$ given the observation $\mathbf{z}_t$ and control $\mathbf{u}_t$.

The states of the hand-object system are $\mathbf{x}_t = (\theta_t, \mathbf{r}_t, \mathbf{t}_t)$. Using those pose parameters (along with the known object model and shape parameters β), we can reconstruct the entire scene. For most of the cases in real life and nearly all of the cases in existing datasets, hands can be modelled as actuators whereas objects are primarily passive. Therefore, the control input for the system is the hand pose $\mathbf{u}_t = \theta$.

Since the differentiable renderer can output various quantities (2D keypoints, bounding box, full image set), the observations we can choose are also flexible. For instance, the 2D positions of the fingertips can serve as the only feedback supervision. Then the observation function is,

$$h(\mathbf{x}_t) = S_{tips} \cdot \mathbf{V}_{hand}(\theta_t), \tag{8}$$



(a) Input (b) Semi [5] (c) HANDYPRIORS

Fig. 4: **Real-world pose estimation.** (a) We record a robotic hand manipulating a cracker box in the real world using a calibrated camera. With (b) initialization using [5]. We observe that (c) our method can recover the transformation of the box.

where $S_{tips} \in \mathbb{R}^{5\times 778}$ is a one-hot encoding matrix that selects fingertips out of the hand.

The more challenging part is to construct the dynamics function $f(\cdot)$. A direct solution is to use a differentiable full dynamics model to describe the system. However, the actuation signals of the hand (like muscle contraction) are unavailable from the MANO parameter. We thus choose to use simpler and more robust quasistatic dynamics. If the hand makes contact with an object on vertices $\mathbf{v}_{t-1} = S_{contact} \cdot \mathbf{V}_{hand}(\theta_{t-1})$ at time step $t-1$, we assume the relative sliding on the vertices from $\mathbf{v}_{t-1}$ to $\mathbf{v}_t \in \mathbb{R}^{n_{contact}\times 3}$ is small. The rotation $\mathbf{R}$ and translation $\mathbf{T}$ are estimated as,

$$\begin{aligned} \mathbf{U}\Sigma\mathbf{V}^\dagger &= svd(\tilde{\mathbf{v}}_{t-1}^\top \tilde{\mathbf{v}}_t) \\ \mathbf{R}_t &= (\mathbf{U}\mathbf{V}^\dagger)^\top \mathbf{R}_{t-1} \\ \mathbf{T}_t &= \bar{\mathbf{v}}_t - \bar{\mathbf{v}}_{t-1}(\mathbf{U}\mathbf{V}^\dagger) + \mathbf{T}_{t-1} \end{aligned} \tag{9}$$

where $\tilde{\mathbf{v}} = \mathbf{v} - \bar{\mathbf{v}}$ is the contact vertices subtracting their centroids $\bar{\mathbf{v}}$, $svd(\cdot)$ is the singular value decomposition.

With the differentiable observation $h(\cdot)$ (Eq. 8) and dynamics $f(\cdot)$ (Eq. 9), we can then use the EKF to track and estimate the states $\mathbf{x}_t = (\theta_t, \mathbf{r}_t, \mathbf{t}_t)$.

IV. EXPERIMENTS

In this section, we apply our differentiable priors to five scenarios: (a) object pose estimation for robotic manipulation, (b) pose estimation, (c) filtering-based tracking, (d) contact refinement, and (e) human-object pose estimation. Our method can achieve comparable or better results in a wide range of tasks.

A. Real-world Robotic Hand and Object Iteration

Our method can generalize to a wide range of scenarios, including robotic hands and object interaction. To demonstrate this, we designed a robotic hand manipulation task. We first record sequences of RGB images where an Allegro robotic hand interacts with a YCB object [57] using the calibrated Intel RealSense camera and record the corresponding hand pose parameters. We then predict the initial object pose using only the RGB image with [5]. Given the robotic hand configuration calculated using the hand pose parameter and segmentation masks predicted by Pointrend [55]. We then use our pipeline to optimize the object pose $\mathbf{r}, \mathbf{t}$. Figure 4 shows the optimization result of our method. The results demonstrate that our method is capable of generating accurate object poses even with robotic hand interaction scenes.

TABLE II: **Evaluation of hand pose estimation.** We measure the joint error (cm), mesh error (cm), and joint AUC on the DexYCB dataset [38]. PA is Procrustes Alignment.

Methods	Joint (PA)	Joint	Mesh (PA)	Mesh	Joints AUC
Artiboost [40]	0.65	1.12	0.69	1.21	0.870
Homan [7]	1.84	4.41	1.80	4.34	0.644
HandOccNet [41]	0.62	1.16	0.66	1.20	0.877
HANDYPRIORS (ours)	**0.52**	**0.81**	**0.51**	**0.89**	**0.896**

TABLE III: **Quantitative evaluation of object pose estimation**. Our approach is compared with Semi [5], Artiboost [40], Homan [7] on the HO3D [11] and DexYCB Dataset [38]. We report the 2D (cm) and 3D (cm) errors.

Method	2D Error (HO3D)	3D Error (HO3D)	2D Error (DexYCB)	3D Error (DexYCB)
Artiboost [40]	6.41	10.83	3.01	6.78
Homan [7]	4.60	41.33	15.21	95.07
Semi [5]	2.06	26.35	9.20	47.77
HANDYPRIORS (ours)	**1.67**	**3.22**	**0.96**	**3.79**

B. Pose Estimation

1) Datasets and Metrics: **Datasets**. HO3D dataset [11] is a real hand-object interaction dataset that contains approximately 80K images from more than 65 sequences. It uses 10 models from the YCB dataset [57]. We report the performance of all of the methods on the evaluation split of the dataset. Dex-YCB dataset [38] consists of 582K images over 1,000 sequences of 10 subjects grasping 20 different YCB objects [57] from 8 camera views. In this work, we report the performance of our method on the S_0 test split of the dataset. Following [40], we filter out frames that contain the left hand or where the hand is not in contact with the object. **Metrics**. In Table III, we measure the 2D error by projecting the object vertices onto the image plane and computing the Chamfer distance between the ground truth vertices and predicted vertices. The 3D error is the Chamfer distance in the original 3D space. We also report the collision between the hand and the object. In Table II, we compute the joint error, mesh error, and joint AUC (Area Under the ROC Curve). For joint and mesh error, we report results before and after Procrustes alignment, where Procrustes alignment changes the global rotation, translation, and scale of the estimated pose to match the ground truth.

2) Optimization-based Refinement: We compare our optimization-based method described in Section III-D with Semi [5], Artiboost [40], Homan [7] for the pose estimation task. In the HO3D dataset, we test three objects from the evaluation split: Mustard Bottle (SM11), Meat Can (MPM10-15), and Bleach Cleanser (SB11, 13). Table III shows that our method outperforms alternative methods for estimating the pose of the objects. Our method also achieves comparable results in terms of collision metrics. Table II shows the hand prediction results compared to Artiboost [40], Homan [7], and HandOccNet [41]. Our method also improves over the initialization from [41] and achieves the best accuracy in all metrics. Figure 5 shows some qualitative results of the predictions. Compared to other methods, our optimized results are better aligned with the image and have more reasonable relative positions between hands and objects.

3) Ablation Study: In this part, we study the influence of different loss terms on the estimation results. All the experiments are run in the evaluation set SM1 from the HO3D dataset [11]. Table I presents the quantitative results. Starting from the initialization by the pre-trained network [5], adding rendering loss $\mathcal{L}_{rendering}$ drastically improves the performance, especially with regards to the 3D error as it can adjust the translation errors in the depth direction. Moreover, adding the physics loss term $\mathcal{L}_{physics}$ improves the results further by reducing penetration and oscillation in the predicted sequences.

TABLE IV: **Error after contact-based pose refinement.** Our differentiable contact module achieves comparable results as ContactOpt [42].

Method	hand joints	hand verts	object verts
ContactOpt	0.0143	0.0137	0.0367
ours	0.0143	0.0136	0.0364

C. Filtering-based tracking

We apply the tracking algorithm described in Section III-E to perform pose estimation. We run on the SM1 sequence and qualitative results are shown in Figure 6. Our filtering pipeline (b) yields adequate estimation results of the objects given only the fingertips as inputs (red dots in the figures). Meanwhile, the (a) learning-based method [5] would have huge errors in object pose estimation.

We also conduct quantitative experiments in Table I where different observations are fed into the filter. $\mathcal{O}_{hand}$ use the predicted 3D positions of fingertips from as observation; $\mathcal{O}_{object}$ takes the predicted object center as observations; $\mathcal{O}_{hand+object}$ uses both the aforementioned hand and object information as an observation. The results show that with more observation available, the tracking results tend to be more accurate. The tracking method is much faster than the optimization pipeline; however, the accuracy decreases since less information is provided to it.

D. Contact Refinement

The physics module can output contact information during prediction. Given the pose of the hand and the object, our GPU-based differentiable Warp kernel [59] can output the signed distance function s and the normal direction $\mathbf{n}$ for each hand vertex $\{\mathbf{n}_i, s_i\}_{i<n_{hand}} = contact(\theta, \mathbf{r}, \mathbf{t})$. Following [42], we define a contact value $CO_i = \min(\frac{1}{s_i}, 1)$ for each vertex. When a vertex is within 1mm ($s_i < 1$) of the object, $CO_i = 1$ then means this point is in-contact.

Given randomly initialized hand poses, our method refers to the ground truth contact information provided by ContactPose datasets [60] as the target. Since our $contact(\cdot)$ module is also differentiable, we can then optimize the poses to match the target contact status $\{CO_i, \mathbf{n}_i\}$ with gradient-based optimization and achieve comparable results as [42] shown in Figure IV.

Figure 7 shows examples of the optimization process. (a) and (c) are the input randomized hand, (b) and (d) are the optimized hand-object interaction. The results are similar to how people handle those objects in daily life.

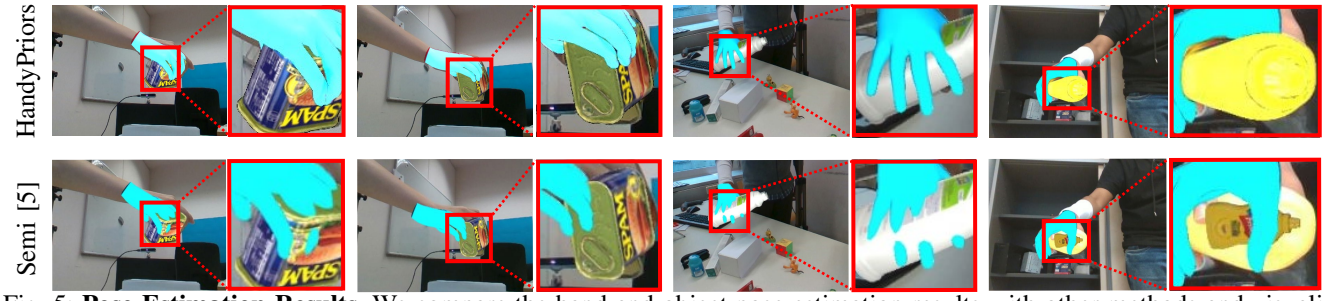
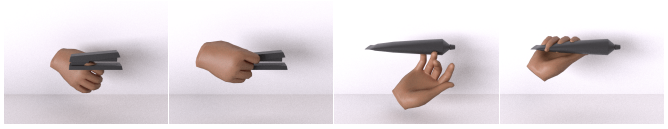


Fig. 5: **Pose Estimation Results.** We compare the hand and object pose estimation results with other methods and visualize the 3D meshes projected on the original image. Our method yields reasonable joint estimation for the interaction scenes.

(a) Semi [5] (b) Filtering (c) Semi [5] (d) Filtering

Fig. 6: **Filtering results.** We show that the filtering pipeline can estimate the object pose with just fingertips (red points) as input. It can give robust and continuous estimation even when the purely learning-based method (a, c) fails.



(a) Initialized (b) Optimized (c) Initialized (d) Optimized

Fig. 7: **Qualitative results of contact-based pose refinement.** Given randomized initial poses (a,c) and desired contact regions, our method can refine poses using a differentiable contact model & gradient-based optimization (b,d).

E. Generalization to Human

Our proposed differentiable priors can also be applied to human object pose estimation. With RGB images from the COCO2017 dataset [56], we use [53] to initialize both human and object poses, and then use [55] to predict the instance segmentation mask for both the human and objects. Subsequently, we replaced the MANO hand model [20] in our pipeline by SMPL human body model [61] which describes the human body also by pose $\theta \in \mathbb{R}^{72}$ and shape $\beta \in \mathbb{R}^{10}$ parameters. Following the same set up as hand-object optimization, we refine human parameters $(\theta, \beta, T_{human})$ and object parameters$(\mathbf{r}, \mathbf{t})$ through our pipeline using both differentiable rendering and physics priors. Our experiment shows that our method can generate accurate human-object pose estimation. As ground truth poses for human and object are unavailable, we report the intersection of union(IoU) and the interpenetration volume between human and object in Table V. Figure 8 shows the qualitative result of our method and the contribution of each component of the pipeline.

V. Conclusion

In this paper, we present HandyPriors, a framework of differentiable rendering and physics priors for pose estimation in hand-object-interaction scenes. Two alternatives are presented to match different use cases: the optimization-based pose refinement which focuses on the prediction accuracy, and the filtering-based tracking which focuses on the runtime speed. Our proposed rendering and physics-related operators enable closed-loop optimization and tracking with self-supervision from the input RGB image sequences. Experiments show that our method yields comparable or better results in the pose estimation and contact refinement task due to our unified and general priors. We also demonstrate that our method can be applied to robotic hand manipulation and human pose estimation tasks.

There could be several future follow-ups to this work. First, the optimization pipeline can fail if the initialization is poor. This problem could be alleviated by adding bounding boxes or key-point related terms to the differentiable rendering module. Second, the performance of the optimization could be improved through a faster differentiable rendering module, such as Nvdiffrast [62]. Third, the current filtering pipeline based on EKF could potentially output the uncertainty of the prediction. It would be promising to incorporate our pipeline with probabilistic prediction. Fourth, while it currently supports human hands, it could be adapted for robotic arms with different configurations by changing the kinematic structures.

TABLE V: **Quantitative evaluation on human object pose estimation** Our method can achieve higher 2D IoU and lower collision error (cm^3) compared to PHOSA [53].

Method	human IoU	object IoU	Collision
PHOSA	0.45	0.51	6818.18
+$\mathcal{L}_{render}$	0.64	0.59	8742.63
+$\mathcal{L}_{physics}$	**0.68**	**0.64**	**666.67**

(a) Initialization (b) $\mathcal{L}_{render}$ (c) $\mathcal{L}_{render+physics}$

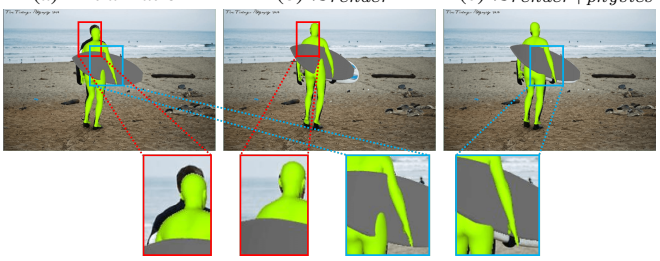


Fig. 8: **Human Object Pose Refinement.** Qualitative comparison between (a) initialization using [53], (b) refinement using only differentiable rendering, and (c) refinement using both differentiable rendering and physics simulator. The rendering loss helps to align the image, and the physics loss can mitigate penetrations.

REFERENCES

[1] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim, "First-person hand action benchmark with rgb-d videos and 3d hand pose annotations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 409–419.

[2] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[3] R. P. Sharma and G. K. Verma, "Human computer interaction using hand gesture," *Procedia Computer Science*, vol. 54, pp. 721–727, 2015.

[4] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid, "Learning joint reconstruction of hands and manipulated objects," in *CVPR*, 2019.

[5] S. Liu, H. Jiang, J. Xu, S. Liu, and X. Wang, "Semi-supervised 3d hand-object poses estimation with interactions in time," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2021.

[6] Z. Cao, I. Radosavovic, A. Kanazawa, and J. Malik, "Reconstructing hand-object interactions in the wild," *ICCV*, 2021.

[7] Y. Hasson, G. Varol, C. Schmid, and I. Laptev, "Towards unconstrained joint hand-object reconstruction from rgb videos," in *2021 International Conference on 3D Vision (3DV)*, 2021.

[8] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 000–16 009.

[9] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[10] M. I. Ribeiro, "Kalman and extended kalman filters: Concept, derivation and properties," *Institute for Systems and Robotics*, vol. 43, no. 46, pp. 3736–3741, 2004.

[11] S. Hampali, M. Rad, M. Oberweger, and V. Lepetit, "Honnotate: A method for 3d annotation of hand and object poses," in *CVPR*, 2020.

[12] S. Hampali, S. D. Sarkar, M. Rad, and V. Lepetit, "Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation," in *CVPR*, 2022.

[13] G. Billings and M. Johnson-Roberson, "Silhonet: An rgb method for 6d object pose estimation," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3727–3734, 2019.

[14] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, "Learning 6d object pose estimation using 3d object coordinates," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 536–551.

[15] Y. Hu, J. Hugonot, P. Fua, and M. Salzmann, "Segmentation-driven 6d object pose estimation," in *CVPR*, 2019.

[16] B. Tekin, S. N. Sinha, and P. Fua, "Real-Time Seamless Single Shot 6D Object Pose Prediction," in *CVPR*, 2018.

[17] Y. He, H. Huang, H. Fan, Q. Chen, and J. Sun, "Ffb6d: A full flow bidirectional fusion network for 6d pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[18] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion," in *Computer Vision and Pattern Recognition (CVPR)*, 2019.

[19] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," in *Robotics: Science and Systems (RSS)*, 2018.

[20] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 2017.

[21] Y. Cai, L. Ge, J. Cai, and J. Yuan, "Weakly-supervised 3d hand pose estimation from monocular rgb images," in *European Conference on Computer Vision*, 2018.

[22] L. Ge, Y. Cai, J. Weng, and J. Yuan, "Hand pointnet: 3d hand pose estimation using point sets," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[23] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt, "Ganerated hands for real-time 3d hand tracking from monocular rgb," in *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2018.

[24] A. Spurr, J. Song, S. Park, and O. Hilliges, "Cross-modal deep variational hand pose estimation," in *CVPR*, 2018.

[25] C. Zimmermann and T. Brox, "Learning to estimate 3d hand pose from single rgb images," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[26] L. Yang and A. Yao, "Disentangling latent hands for image synthesis and pose estimation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[27] S. Yuan, G. Garcia-Hernando, B. Stenger, G. Moon, J. Chang, K. Lee, P. Molchanov, J. Kautz, S. Honari, L. Ge, J. Yuan, X. Chen, G. Wang, F. Yang, K. Akiyama, Y. Wu, Q. Wan, M. Madadi, S. Escalera, S. Li, D. Lee, I. Oikonomidis, A. Argyros, and T. Kim, "Depth-based 3d hand pose estimation: From current achievements to future goals," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[28] S. Baek, K. Kim, and T. Kim, "Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[29] A. Boukhayma, R. d. Bem, and P. H. Torr, "3d hand shape and pose from images in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 843–10 852.

[30] Y. Rong, T. Shiratori, and H. Joo, "Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration," in *IEEE International Conference on Computer Vision Workshops*, 2021.

[31] C. Zimmermann, D. Ceylan, J. Yang, B. Russel, M. Argus, and T. Brox, "Freihand: A dataset for markerless capture of hand pose and shape from single rgb images," in *IEEE International Conference on Computer Vision (ICCV)*, 2019. [Online]. Available: "https://lmb.informatik.uni-freiburg.de/projects/freihand/"

[32] D. Kulon, R. A. Guler, I. Kokkinos, M. M. Bronstein, and S. Zafeiriou, "Weakly-supervised mesh-convolutional hand reconstruction in the wild," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[33] G. Moon, S.-I. Yu, H. Wen, T. Shiratori, and K. M. Lee, "Interhand2.6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image," in *European Conference on Computer Vision (ECCV)*, 2020.

[34] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt, "Real-time hand tracking under occlusion from an egocentric rgb-d sensor," in *Proceedings of International Conference on Computer Vision (ICCV)*, 2017. [Online]. Available: http://handtracker.mpi-inf.mpg.de/projects/OccludedHands/

[35] P. Panteleris, I. Oikonomidis, and A. Argyros, "Using a single rgb frame for real time 3d hand pose estimation in the wild," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.

[36] Y. Liu, Y. Liu, C. Jiang, K. Lyu, W. Wan, H. Shen, B. Liang, Z. Fu, H. Wang, and L. Yi, "Hoi4d: A 4d egocentric dataset for category-level human-object interaction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 21 013–21 022.

[37] S. Brahmbhatt, C. Tang, C. D. Twigg, C. C. Kemp, and J. Hays, "ContactPose: A dataset of grasps with object contact and hand pose," in *The European Conference on Computer Vision (ECCV)*, August 2020.

[38] Y.-W. Chao, W. Yang, Y. Xiang, P. Molchanov, A. Handa, J. Tremblay, Y. S. Narang, K. Van Wyk, U. Iqbal, S. Birchfield, J. Kautz, and D. Fox, "DexYCB: A benchmark for capturing hand grasping of objects," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[39] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim, "First-person hand action benchmark with rgb-d videos and 3d hand pose annotations," in *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2018.

[40] L. Yang, K. Li, X. Zhan, J. Lv, W. Xu, J. Li, and C. Lu, "ArtiBoost: Boosting articulated 3d hand-object pose estimation via online exploration and synthesis," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[41] J. Park, Y. Oh, G. Moon, H. Choi, and K. M. Lee, "Handoccnet: Occlusion-robust 3d hand mesh estimation network," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[42] P. Grady, C. Tang, C. D. Twigg, M. Vo, S. Brahmbhatt, and C. C. Kemp, "Contactopt: Optimizing contact to improve grasps," in *Pro-*

ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1471–1481.
[43] H. Hu, X. Yi, H. Zhang, J.-H. Yong, and F. Xu, "Physical interaction: Reconstructing hand-object interactions with physics," in *SIGGRAPH Asia 2022 Conference Papers*, 2022.
[44] S. Liu, T. Li, W. Chen, and H. Li, "Soft rasterizer: A differentiable renderer for image-based 3d reasoning," *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019.
[45] S. Laine, J. Hellsten, T. Karras, Y. Seol, J. Lehtinen, and T. Aila, "Modular primitives for high-performance differentiable rendering," *ACM Trans. Graph.*, vol. 39, no. 6, 2020.
[46] M. Nimier-David, D. Vicini, T. Zeltner, and W. Jakob, "Mitsuba 2: A retargetable forward and inverse renderer," *ACM Trans. Graph.*, vol. 38, no. 6, 2019.
[47] E. Heiden, D. Millard, E. Coumans, Y. Sheng, and G. S. Sukhatme, "NeuralSim: Augmenting differentiable simulators with neural networks," *arXiv:2011.04217*, 2020.
[48] J. Xu, T. Chen, L. Zlokapa, M. Foshey, W. Matusik, S. Sueda, and P. Agrawal, "An end-to-end differentiable framework for contact-aware robot design," *arXiv preprint arXiv:2107.07501*, 2021.
[49] E. Heiden, C. E. Denniston, D. Millard, F. Ramos, and G. S. Sukhatme, "Probabilistic inference of simulation parameters via parallel differentiable simulation," *International Conference on Robotics and Automation (ICRA)*, 2022.
[50] S. Christen, M. Kocabas, E. Aksan, J. Hwangbo, J. Song, and O. Hilliges, "D-grasp: Physically plausible dynamic grasp synthesis for hand-object interactions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
[51] D. Turpin, L. Wang, E. Heiden, Y.-C. Chen, M. Macklin, S. Tsogkas, S. Dickinson, and A. Garg, "Grasp'd: Differentiable contact-rich grasp synthesis for multi-fingered hands," in *European Conference on Computer Vision (ECCV)*, 2022.
[52] S. K. Dwivedi, N. Athanasiou, M. Kocabas, and M. J. Black, "Learning to regress bodies from images using differentiable semantic rendering," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 250–11 259.
[53] J. Y. Zhang, S. Pepose, H. Joo, D. Ramanan, J. Malik, and A. Kanazawa, "Perceiving 3d human-object spatial arrangements from a single image in the wild," in *European Conference on Computer Vision (ECCV)*, 2020.
[54] E. Gärtner, M. Andriluka, E. Coumans, and C. Sminchisescu, "Differentiable dynamics for articulated 3d human motion reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 190–13 200.
[55] A. Kirillov, Y. Wu, K. He, and R. Girshick, "Pointrend: Image segmentation as rendering," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
[56] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision (ECCV)*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., 2014.
[57] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *2015 International Conference on Advanced Robotics (ICAR)*, 2015.
[58] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 2014.
[59] M. Macklin, "Warp: A high-performance python framework for gpu simulation and graphics," https://github.com/nvidia/warp, March 2022, nVIDIA GPU Technology Conference (GTC).
[60] S. Brahmbhatt, C. Tang, C. D. Twigg, C. C. Kemp, and J. Hays, "Contactpose: A dataset of grasps with object contact and hand pose," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*. Springer, 2020, pp. 361–378.
[61] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 2015.
[62] S. Laine, J. Hellsten, T. Karras, Y. Seol, J. Lehtinen, and T. Aila, "Modular primitives for high-performance differentiable rendering," *ACM Transactions on Graphics*, vol. 39, no. 6, 2020.