Novel-view Synthesis and Pose Estimation for Hand-Object Interaction from Sparse Views

Wentian Qu[1,2] Zhaopeng Cui[3] Yinda Zhang[4] Chenyu Meng[1,2] Cuixia Ma[1,2]
Xiaoming Deng[1,2*] Hongan Wang[1,2*]
[1]Institute of Software, Chinese Academy of Sciences [2]University of Chinese Academy of Sciences
[3]State Key Lab of CAD&CG, Zhejiang University [4]Google

Abstract

Hand-object interaction understanding and the barely addressed novel view synthesis are highly desired in the immersive communication, whereas it is challenging due to the high deformation of hand and heavy occlusions between hand and object. In this paper, we propose a neural rendering and pose estimation system for hand-object interaction from sparse views, which can also enable 3D hand-object interaction editing. We share the inspiration from recent scene understanding work that shows a scene specific model built beforehand can significantly improve and unblock vision tasks especially when inputs are sparse, and extend it to the dynamic hand-object interaction scenario and propose to solve the problem in two stages. We first learn the shape and appearance prior knowledge of hands and objects separately with the neural representation at the offline stage. During the online stage, we design a rendering-based joint model fitting framework to understand the dynamic hand-object interaction with the pre-built hand and object models as well as interaction priors, which thereby overcomes penetration and separation issues between hand and object and also enables novel view synthesis. In order to get stable contact during the hand-object interaction process in a sequence, we propose a stable contact loss to make the contact region to be consistent. Experiments demonstrate that our method outperforms the state-of-the-art methods. Code and dataset are available in project webpage https://iscas3dv.github.io/HO-NeRF.

1. Introduction

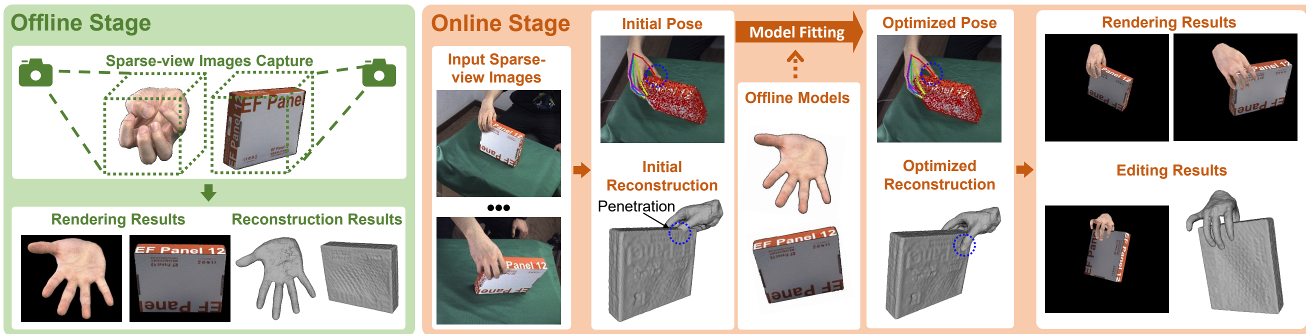
Hand-object interaction understanding plays an important role in immersive contextual teaching applications such as surgical operation and training in the use of machinery. Previous works mostly focus on the hand-object interaction detection [10], reasoning [26] or pose estimation [17, 16]. However, the barely addressed novel view synthesis of hand-object interaction is also highly desired.

Recently, neural rendering is emerging to facilitate the novel view synthesis simply by learning from a collection of images and produces promising high-quality images. Although existing neural rendering approaches perform well on static scenes [31, 2], rigid objects [54, 12] and human models [39, 38, 45], they barely considered scene context in interaction (such as contact [63] and model penetration [4, 24]). In the realm of hand, LISA [8] is the only hand neural rendering model, and achieves promising rendering results of bare hands. However, LISA cannot work well for hand-object interaction due to heavy inter-occlusions and it requires dense (about 20) camera views that may refrain it from wide applications. It is even more challenging to use sparse-view images to synthesize novel views [49, 25] and estimate accurate pose for hand-object interaction, which plays a key role in many applications such as Holoportation [36] and manipulation skill learning from human demonstration [42, 1].

In this work, we propose a novel-view synthesis and pose estimation system for hand-object interaction scenes with sparse camera views (Fig. 1). Recent scene understanding work [53] shows a scene specific model built beforehand can significantly improve and unblock vision tasks especially when inputs are sparse, and we extend it from static objects to dynamic hand-object interaction scenes and solve the problem in two stages. We first use sparse-view images as input to train the pose-driven neural rendering models of hand and object during the offline stage. Benefiting from the progress of hand pose tracking [15, 19, 28] and object pose estimation [27], we only need very low cost to build hand model and object model. Then at the online stage, we estimate both hand and object poses using a novel differentiable rendering-based model fitting under geometric constraints. In this way, we can understand hand-object interaction accurately and render novel views effectively.

However, it is non-trivial to fulfill this goal. Firstly, it is difficult to build neural rendering systems from sparse cam-

*indicates corresponding author.



(a) Pre-built hand and object models (b) Online initialization (c) Online fitting and editing

Figure 1: We propose a neural rendering and pose estimation system for hand-object interaction using sparse view images. (a) During offline stage, we learn hand and object models that enable rendering and shape reconstruction. During online stage, we initialize the pose from sparse camera views (b), and then conduct online fitting to improve pose estimation, which enables photo-realistic free viewpoint rendering (c). Our framework also naturally supports hand object interaction editing.

era views due to insufficient visual information and depth ambiguity caused by hand-object inter-occlusions. Existing few-shot neural rendering methods [25, 49] fail under sparse camera views (Fig. 7). In order to solve this problem, we establish the fitting process based on the pre-built models which provides strong shape and appearance priors, and it can achieve excellent novel view rendering from sparse views. Secondly, it is difficult to obtain accurate hand and object poses and reasonable interactions only by photometric constraints due to extensive occlusion. In order to handle this problem, we propose a novel differentiable rendering-based model fitting process under geometric constraints to refine the poses and enforce the spatial context between hand and object by leveraging the signed distance function (SDF) to reduce penetration and to encourage tight hand-object regions to contact. We propose a stable contact loss to penalize large sliding of the hand-object contact area across temporally consecutive frames. Through the joint model fitting process, we can achieve accurate pose estimation for hand-object interactions. Thirdly, our system needs a dataset to include images of hand, object, and hand-object interaction. However, existing datasets such as [5, 13] cannot satisfy this requirement, so we need to collect a real dataset to evaluate our method.

Our main contribution is summarized as follows. First, to the best of our knowledge, we present the first solution to unblock hand-object interaction neural rendering from sparse views. We design a new two-stage approach (i.e. offline model building and online model fitting) to achieve accurate hand-object pose estimation and photo-realistic novel view synthesis. Second, we leverage effective geometric constraints to conduct rendering-based model fitting, which can recover reasonable hand-object interaction even under sparse views and inter-occlusions. To reduce the sliding that occurs during the interaction, we design a new stable contact loss to enforce the hand-object contacted regions to be consistent for video sequences. Third, we propose a hand-object interaction dataset *HandObject* for neural rendering tasks, including images for hand, object and hand-object interaction scenes. Finally, experiments demonstrate that, with the help of offline and online stages, our method achieves significantly better performance in pose estimation and rendering quality than previous methods.

2. Related Work

Hand-Object Interaction Understanding. Existing hand-object interaction understanding works [4, 21, 55, 61, 62, 16, 26, 13, 17, 24, 59, 14, 46, 47, 7, 65] usually use hand statistical shape model such as MANO [44], and adopt known object shape to estimate object 6D pose. The geometric feature extracted from implicit network can also be used to represent object shape [59, 46]. The key challenges of hand-object understanding include occlusion, penetration and separation between hand and object. However, mesh-based hand and object representation is expensive for surface-based penetration and contact loss [17]. In order to deal with the contact for hand-object interaction understanding, several existing works [17, 55, 21, 62, 24] achieve reasonable contact by predicting hand regions where contact is likely to occur and optimizing the distance between the object vertices and contact regions on the hand. Recently, implicit shape representations such as SDF [24, 4, 59, 7] are emerged to facilitate the detection of the penetration and contact between hand and object, because SDF can indicate the spatial relationship between point and surface and the penetration of two shapes can be judged with the sign of SDF values. Our method adopts SDF representation to facilitate geometric constraints in hand-object interaction.

Neural Rendering. Neural radiance fields (NeRF) [31] aims to synthesize novel view of a scene via volume ren-

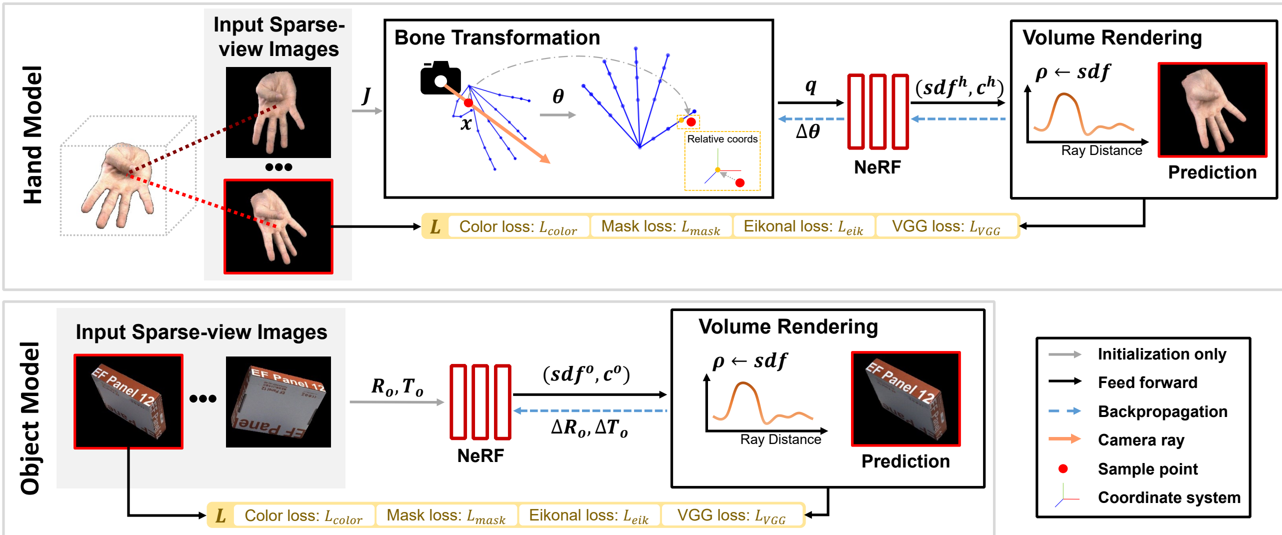


Figure 2: Offline stage to learn hand and object models. Top: Hand model. Each sampling point on the ray is converted to the local coordinate system of each hand part through bone transformation. Then we encode the point into embedding vectors and feed it to hand model to get the SDF and color value. Bottom: Object model. We convert the sampling point to the model coordinate with the object pose, and get the SDF and color value.

dering with densely sampled posed images. The few-shot novel view synthesis methods [60, 49, 25, 20] use sparse view images as input to build neural radiation fields, but fail in widely divergent camera view. Although the shape surface of an object is implicitly included in NeRF, the traditional density cannot extract accurate surface [48]. Therefore, IDR [58], NeuS [48] and VolSDF [57] use SDF or occupancy fields combined with volume rendering to achieve high precision reconstruction. But these methods only reconstruct rigid objects. From the perspective of new scene synthesis, several NeRF editing works [40, 37, 54, 34] has been proposed. However, these methods cannot be trivially extended to edit non-rigid objects such as human hand.

Neural Articulated Shape Representation. Prior arts [9, 30, 6, 23, 39, 38, 35] use implicitly shape representation to reconstruct articulated human body shape. NASA [9] and its variants [30, 6] generates human body shape by converting the posed human body to the canonical pose and querying the SDF value of a point in the canonical pose space. In order to learn generative novel view synthesis, several methods [39, 38, 35, 45, 51, 64, 50, 8] integrate bone transformation [45] and linear blend skinning [38, 64, 50, 8] with neural radiance fields. Different from the neural rendering systems [39, 35, 45], we present a hand-object interaction neural rendering method using sparse-view images, in which spatial context between hand and object are modeled by the SDF representations to reduce penetration and encourage stable contact. Compared to implicit articulated shape representation such as NASA [9], Animatable NeRF [38] and DD-NeRF [56] that need parametric shape models or skinning weight supervision, our method can learn the geometry and appearance of hand and object with sparse-view only.

3. Method

Given sparse-view observations of hand-object interaction, we aim to generate free-viewpoint synthesis of the scene and estimate hand skeleton pose and object 6D pose. Our framework is divided into two stages: offline model building and online model fitting. At the offline stage, we learn the neural models for hand and object individually based on the pre-captured sparse-view images (Sec. 3.1). As shown in Fig. 2, our neural hand model is a generative implicit representation driven by hand skeleton pose, which can be used to represent geometry and to generate novel views and our object model can be driven by object 6D pose. At the online stage (Sec. 3.2), given the sparse-view images, we estimate both hand and object poses using a rendering-based model fitting under effective geometric constraints (Fig. 3). For video input, we can further enforce smooth and stable contact loss to reduce pose jitters between frames to generate more smooth and consistent hand-object interaction. Benefited from our offline models and online fitting method, we can also edit the hand-object interaction scenes (Sec. 4.5).

3.1. Offline Stage for Hand-Object Model Building

Hand Neural Rendering Model. We aim to build a pose-driven hand model that can achieve novel view synthesis

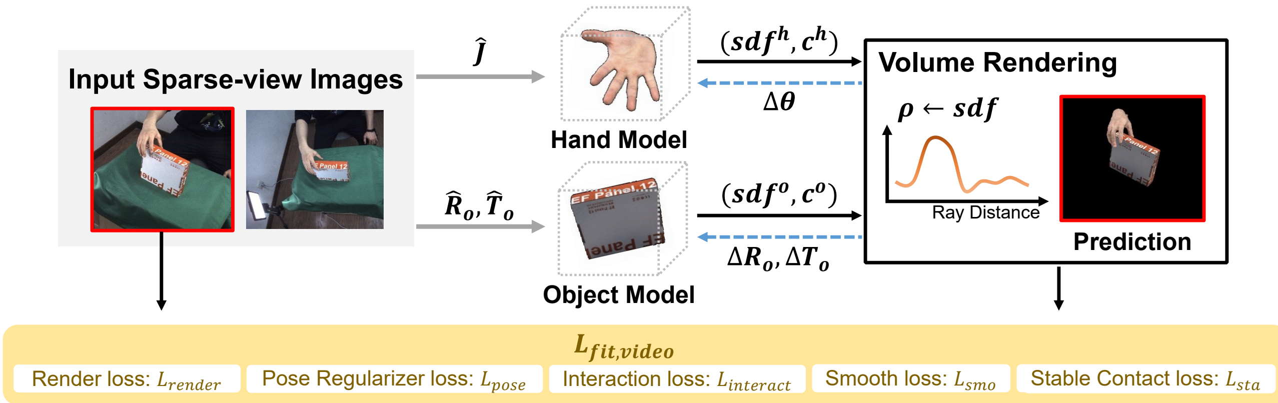


Figure 3: Online stage for joint model fitting. We utilize hand/object pose estimation networks for initialization, and refine pose with $\mathcal{L}_{fit}$ for single frame and $\mathcal{L}_{fit,video}$ for video sequence.

and recover accurate geometry (Top of Fig. 2). In our hand model, we convert sampling point on camera ray to local coordinate system of each hand part through bone transformation. Then we encode the point into embedding vectors and feed it to hand model to get the SDF and color value. We use SDF-based implicit fields for hand geometry representation [48], because it enables more accurate surface than NeRF's density fields and facilitates geometric constraints during model fitting (Sec. 3.2). Our hand model can be formulated as:

$$f^s_{hand}(\mathbf{x}(t), \mathbf{J}) = sdf^h, F^h, \\ f^c_{hand}(\mathbf{x}(t), \mathbf{J}, F^h, \mathbf{n}_h) = \mathbf{c}^h, \tag{1}$$

where $\mathbf{x}(t) = \mathbf{o} + t\mathbf{d}$ represents the sampling point on the ray, $\mathbf{o} \in \mathbb{R}^3$ is the camera optical center, $\mathbf{d} \in \mathbb{R}^3$ is the ray direction, $sdf^h \in \mathbb{R}$ represents SDF value under hand model, $\mathbf{n}_{hand} \in \mathbb{R}^3$ represents the derivation of sdf^h, $\mathbf{c}^h \in \mathbb{R}^3$ represents the color value, f represents the MLP network and F^h represents the features output by f^s_{hand}. We define the hand skeleton pose as $\mathbf{J} \in \mathbb{R}^{n_j \times 3}$ (n_j=21 is the hand joint number), which can be used to calculate the bone transformation $\mathbf{B}^{-1} \in \mathbb{R}^{n_j \times 4 \times 4}$ and the position of each joint in canonical pose $\mathbf{T} \in \mathbb{R}^{n_j \times 3}$ as used in HALO [23]. Thus each sampling point $\mathbf{x}$ can be converted to the local bone coordinate systems by:

$$\begin{bmatrix}\mathbf{q}\\1\end{bmatrix} = \mathbf{B}^{-1}\begin{bmatrix}\mathbf{x}\\1\end{bmatrix} - \begin{bmatrix}\mathbf{T}\\1\end{bmatrix}. \tag{2}$$

Inspired by A-NeRF [45], we calculate the components of the embedding vectors by $\mathbf{v} = \|\mathbf{q}\|_2$, $h(\mathbf{v}) = 1 - S(\tau(\mathbf{v} - \bar{\mathbf{v}}))$ and $\mathbf{r} = \frac{\mathbf{q}}{\mathbf{v}}$, where $S(\cdot)$ represents the Sigmoid operation, $\bar{\mathbf{v}}$ represents the cutoff point, τ represents the sharpness, and $\gamma(\cdot)$ represents the positional encoding. We combine them as embedding vectors $\mathbf{e}^s_h = [h(v)\gamma(v), h(v)\gamma(\mathbf{r})]$ and feed it into f^s_{hand} to get SDF value. Then we add $\mathbf{n}_{hand}$ in the embedding vector as $\mathbf{e}^c_h = [h(v)\gamma(v), h(v)\gamma(\mathbf{r}), F^h, \gamma(\mathbf{n}_h)]$ and feed it into f^c_{hand} to get color value. Similar to NeuS [48], we use sdf^h to get opaque density ρ^h by:

$$\rho^h(t) = \max\left(\frac{-\frac{\mathrm{d}\Phi_z}{\mathrm{d}t}(f^s_{hand}(\mathbf{x}(t)))}{\Phi_z(f^s_{hand}(\mathbf{x}(t)))}, 0\right), \tag{3}$$

where $\Phi_z(x) = (1 + e^{-zx})^{-1}$ and z is a learnable scalar. More details can be found in the supplementary material.

Object Neural Rendering Model. Our goal is to generate an object model that can be controlled by the object 6D pose (Bottom of Fig. 2). We can use object 6D pose $\mathbf{R}_o \in \mathbb{R}^{3\times3}$ and $\mathbf{T}_o \in \mathbb{R}^3$ to transform the object from the object coordinate to the world coordinate. For each sampling point $\mathbf{x}$, we first convert it to the object model coordinate system with inverse transformation $\mathbf{q}_o = \mathbf{R}_o^{-1}(\mathbf{x} - \mathbf{T}_o)$, then encode $\mathbf{q}_o$ to the embedding vector, and feed it to the object model to get SDF sdf^o and color $\mathbf{c}^o$ values. Our object model can be formulated as:

$$f^s_{obj}(\mathbf{x}(t), \mathbf{R}_o, \mathbf{T}_o) = sdf^o, F^o, \\ f^c_{obj}(\mathbf{x}(t), \mathbf{R}_o, \mathbf{T}_o, F^o, \mathbf{n}_o) = \mathbf{c}^o. \tag{4}$$

The ray direction under object coordinate system can be expressed by $\mathbf{l}_o = \mathbf{R}_o^{-1}\mathbf{d}$, and we define the normal of $\mathbf{x}$ in object model as $\mathbf{n}_o$. The embedding feature vector of f^s_{obj} can be formulated as: $e^s_o = [\gamma(\mathbf{q}_o)]$ and the embedding feature vector of f^c_{obj} can be formulated as: $e^c_o = [\gamma(\mathbf{q}_o), \gamma(\mathbf{l}_o), F^o, \gamma(\mathbf{n}_o)]$.

Loss Function in Offline Stage. To learn hand and object models, we mainly use color loss and mask loss to encourage rendered color $\hat{C}$ and mask $\hat{M}$ to be closed to the ground truth respectively. We also use Eikonal loss [11] to regularize the SDF and follow [29] to use VGG loss. Therefore, the total loss of hand model can be formulated as:

$$\mathcal{L} = \lambda_{co}\mathcal{L}_{color} + \lambda_m\mathcal{L}_{mask} + \lambda_e\mathcal{L}_{eik} + \lambda_v\mathcal{L}_{VGG}, \tag{5}$$

where $\mathcal{L}_{color}$, $\mathcal{L}_{mask}$ and $\mathcal{L}_{eik}$ are similar to NeuS [48], λ_{co},λ_m,λ_e and λ_v are loss weights.

3.2. Online Stage for Joint Model Fitting

3.2.1 Compositive Volume Rendering

Through the offline stage, we use the shape and appearance priors of hands and objects to build neural models, and then we fix the parameters of these models and optimize the poses of hands and objects in the hand-object interaction scene at online stage (Fig. 3). Given sparse-view images, we first use multi-view-based pose estimation methods [19, 3, 27] to obtain hand and object poses as initialization. For each sampling point $\mathbf{x}$, it passes through hand and object models with initial poses to obtain $[\rho^h, \mathbf{c}^h]$ for hand and $[\rho^o, \mathbf{c}^o]$ for object, respectively. Then the rendering color and the foreground mask can be defined as:

$$\hat{C} = \sum_{i=1}^{N} (T_i \alpha_i^h \mathbf{c}_i^h + T_i \alpha_i^o \mathbf{c}_i^o), \hat{M} = \sum_{i=1}^{N} (T_i \alpha_i^h + T_i \alpha_i^o),$$
$$T_i = \exp\left(- \sum_{j}^{i-1} (\rho^h(j) + \rho^o(j)) \Delta t_j \right), \tag{6}$$

where $\alpha_i = 1 - \exp(-\rho(i)\Delta t_i)$, Δt is the sampling distance between adjacent points along the ray, and N is the number of sampling points along each ray.

3.2.2 Single-Frame Based Loss Function

We use render loss and pose regularizer loss to stabilize the positions of objects and hands, and adopt the interaction loss via SDF representation to avoid the penetration and encourage tight hand-object regions to be contact. Therefore, the loss function in the joint model fitting on single frame can be formulated as:

$$\mathcal{L}_{fit} = \mathcal{L}_{render} + \mathcal{L}_{pose} + \mathcal{L}_{interact}. \tag{7}$$

Render Loss. The render loss consists of color and mask loss: $\mathcal{L}_{render} = \lambda_{co}\mathcal{L}_{color} + \lambda_m \mathcal{L}_{mask}$.

Pose Regularizer Loss. Inspired by [45], we enforce the refined pose to be similar to the initial pose as: $\mathcal{L}_{pose} = \lambda_h \|\mathbf{J} - \hat{\mathbf{J}}\|_2 + \lambda_o \|\mathbf{V} - \hat{\mathbf{V}}\|_2$, where $\mathbf{J}$ and $\mathbf{V}$ are the refined 3D hand joints and object vertices, the $\hat{\mathbf{J}}$ and $\hat{\mathbf{V}}$ are the estimations of hand poses and object vertices as initialization, and λ_h and λ_o are loss weights.

Interaction Loss. The interaction loss includes penetration loss for solving the penetration and contact loss for forming reasonable contacts, which is defined as $\mathcal{L}_{interact} = \lambda_p \mathcal{L}_p + \lambda_c \mathcal{L}_c$, where $\mathcal{L}_p$ and $\mathcal{L}_c$ are the penetration loss and the contact loss, and λ_p and λ_c are loss weights.

For a sampling point $\mathbf{x}$, we calculate its SDF values for hand model $f^s_{hand}(\mathbf{x})$ and object model $f^s_{obj}(\mathbf{x})$, and penalize the SDF values of points, both the SDF values are negative, to become zero. The penetration loss $\mathcal{L}_p$ can be formulated as: $\mathcal{L}_p = \frac{1}{|N_{in}|} \sum_{\mathbf{x} \in N_{in}} -(f^s_{hand}(\mathbf{x}) + f^s_{obj}(\mathbf{x}))$, where N_{in} represents the points whose SDF values for both hand model and object model are negative.

In order to encourage reasonable contact between hand and object, we use contact loss $\mathcal{L}_c$ to enforce the tight hand-object regions to contact: $\mathcal{L}_c = \frac{1}{|N_c|} \sum_{\mathbf{x} \in N_c} (|f^s_{hand}(\mathbf{x})| + |f^s_{obj}(\mathbf{x})|)$, where N_c represents the points with $|f^s_{hand}(\mathbf{x})| + |f^s_{obj}(\mathbf{x})| < \varepsilon$ (ε is set to 0.01).

3.2.3 Video-Based Loss Function

During online stage, our method can be also used for video sequences. We add smooth loss $\mathcal{L}_{smo}$ to reduce the pose jitters between frames. In order to make the contact area more stable and reduce sliding between hand and object, we propose a new stable contact loss $\mathcal{L}_{sta}$. Therefore, the loss function in the joint model fitting for video sequences can be formulated as:

$$\mathcal{L}_{fit,video} = \mathcal{L}_{fit} + \mathcal{L}_{smo} + \mathcal{L}_{sta}. \tag{8}$$

Smooth Loss. The smooth loss encourages velocity of hand joints and object vertices to change smoothly: $\mathcal{L}_{smo} = \frac{1}{N_t - 1} \sum_{i=1}^{N_t - 1} \mu_h \|\mathbf{J}_{i+1} - \mathbf{J}_i\|_2 + \mu_o \|\mathbf{V}_{i+1} - \mathbf{V}_i\|_2$, where N_t is the frame number, and μ_h and μ_o are loss weights.

Stable Contact Loss. Although the above loss functions are effective to get reasonable hand-object interaction results, there are still potential challenges. For example, the contact loss $\mathcal{L}_c$ can effectively encourage contact for each frame, yet there is sliding between the hand and the object in the contact area. To address this issue, we design a new stable contact loss to ensure reasonable and realistic hand-object contact for input sequences (Fig. 4). We fix mesh models for each object pre-built in offline stage, and extract the initial contact vertices on the mesh, then we penalize the inconsistent contact of the vertices between frames.

For frame i, we collect the initial contact vertices $\mathbf{p}_1^i = \mathbf{R}_o^i \mathbf{x}_n^i + \mathbf{T}_o^i$ on object surface, whose SDF value under the hand model $f^s_{hand}(\mathbf{p}_1^i)$ is negative, where $\mathbf{x}_n^i$ represents the contact vertices in object model coordinate. Then the initial contact vertices are transformed to the other frames using $\mathbf{p}_1^j = \mathbf{R}_o^j \mathbf{x}_n^i + \mathbf{T}_o^j$, and penalize the vertices whose $f^s_{hand}(\mathbf{p}_1^j)$ are positive (i.e. contact to non-contact) using the loss $\mathcal{L}_1 = max(f^s_{hand}(\mathbf{p}_1^j), 0)$. In order to avoid the degeneracy of contact (i.e., a wrongly predicted contact vertex of a frame could make its transformations in other frames to be contacted), we also need to refrain the non-contact object vertices from contacting with hand in other frames. Specifically, we query the nearest non-contact object vertex $d(\mathbf{x}_n^i)$ to each contact point in frame i, where $d(\cdot)$ is a

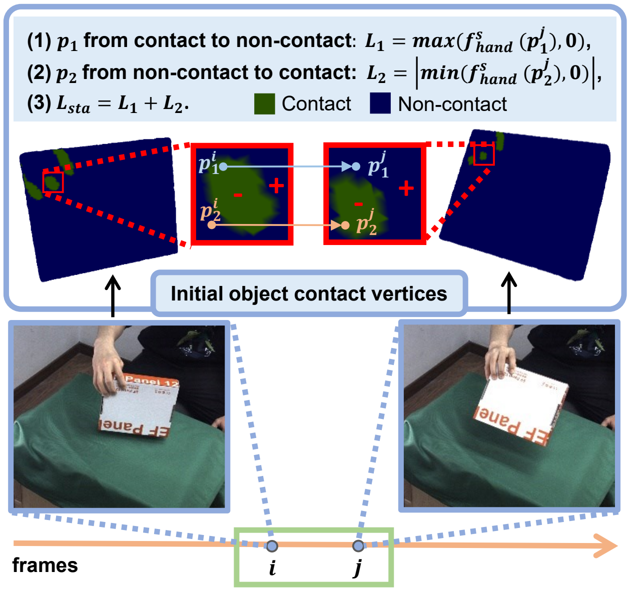


Figure 4: Illustration of the stable contact loss between two frames. We extract the initial contact vertices on the object, and then penalize the inconsistent contact (i.e. contact to non-contact, non-contact to contact) of the object vertices between two frames.

function to locate the closest object vertex to $\mathbf{x}_n^i$ that is not contact with hand. Then transform it to other frames $\mathbf{p}_2^j = \mathbf{R}_o^j d(\mathbf{x}_n^i) + \mathbf{T}_o^j$, and penalize the vertices with negative SDF values for hand model $f_{hand}^s(\mathbf{p}_2^j)$ (i.e. non-contact to contact) using the loss $\mathcal{L}_2 = |min(f_{hand}^s(\mathbf{p}_2^j), 0)|$. Therefore, the stable contact loss can be formulated as:

$$\mathcal{L}_{sta} = \frac{1}{M} \sum_{i=1}^{|N_s|} \sum_{j \neq i}^{|N_s|} (\mu_{si}\mathcal{L}_1 + \mu_{so}\mathcal{L}_2), \quad (9)$$

where N_s represents the frames in which the number of contact points x_n is greater than zero, M is the total number of frame pairs with object contact among the sequence, and μ_{si} and μ_{so} are loss weights.

4. Experiment

4.1. Datasets and Evaluation Metrics

HandObject. Since there is no real-world dataset that can be used to train and evaluate our model, we use a multi-camera system with 8 cameras to collect a hand-object interaction dataset named HandObject. It contains 85k images with the resolution of 266×230.

Synthetic DexYCB. There are no images taken only for hand and object in the original DexYCB [5], which makes it impossible to train our offline model. So we utilize DexYCB to generate a synthetic dataset named Synthetic DexYCB. We use Pytorch3d [43] to render images of 400×400 and use the parametric hand texture model HTML [41] to add texture on hands.

Evaluation Metrics. 1) We use the mean per joint position error (MPJPE) to measure the accuracy of hand skeleton pose. We follow [18] to use the average distance (ADD), average closest point distance (ADD-S) [52] and the value of average distance (AD) for evaluation of object pose estimation. We set ADD and ADD-S to be 15mm. 2) Rendering quality is evaluated with PSNR, SSIM and LPIPS metrics as [31]. 3) We follow [17] to use Penetration depth (mm) and Intersection volume (cm^3) to evaluate the interpenetration level. 4) To evaluate the effectiveness of smooth loss, we use Acceleration error (mm/s^2) [22] to measure the average difference between ground truth 3D acceleration and predicted 3D acceleration of each hand joint (Acc-J) and each object vertex (Acc-V) respectively. To evaluate the effectiveness of our stable contact loss, we use the percentage of contact points IoU (PCI) as a new metric. We fix the mesh model for each object, and calculate PCI by averaging IoU of the contact vertices between adjacent frames.

Implementation Details. During offline stage, we train hand/object models with a single GeForce RTX 3080 Ti, costing 20 hours with 11.0 GB memory and 8 hours with 6.0 GB memory, respectively. We set λ_{co}, λ_m, λ_e to 1, and set λ_v to increment from 0 to 1 after 10k iterations and then keep it at 1 in Eq. 5. We form a training batch by randomly sampling 441 rays from an image, with 64 coarse sampling points and 64 fine sampling points. In the single-frame model fitting part of the online stage (Sec. 3.2), we first use render loss and pose regularizer loss to iterate over each image 30 times, and the loss weights λ_{co}, λ_m, λ_h, and λ_o are set to 1, 0.5, 100, and 5. Then the interaction loss is added and iterates 25 times, where λ_h, λ_o, λ_p, and λ_c are set to 30, 20, 20 and 30. We use the single-frame optimized pose as the initialization, and then add smooth loss and stable contact loss for video-based model fitting. The loss weight $\lambda_{co}, \lambda_m, \mu_h, \mu_o, \mu_{si}, \mu_{so}$ are set to 0.5, 0.25, 50, 50, 100 and 5. During the online stage, we sample 64 coarse sampling points and 128 fine sampling points on a ray. The optimization and rendering times for one frame are 3 minutes and 30 seconds, respectively.

4.2. Novel View Synthesis and Reconstruction

We show novel view synthesis and reconstruction of hand and hand-object interaction in Fig. 5. Our models contain realistic appearance and geometry details and can achieve full 360 degree free-viewpoint rendering.

4.3. Comparison to State-of-the-art Methods

Pose Estimation. In hand pose estimation, we compare with the state-of-the-art (SoTA) multi-view pose estimation

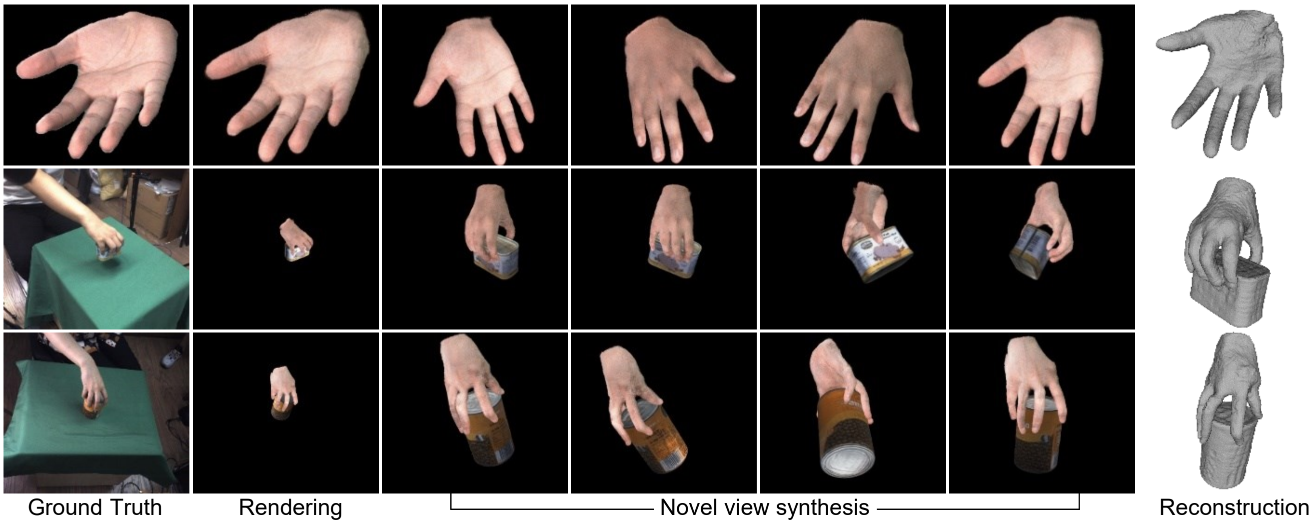


Figure 5: Novel view synthesis and reconstruction of hand and hand-object interaction scenes.

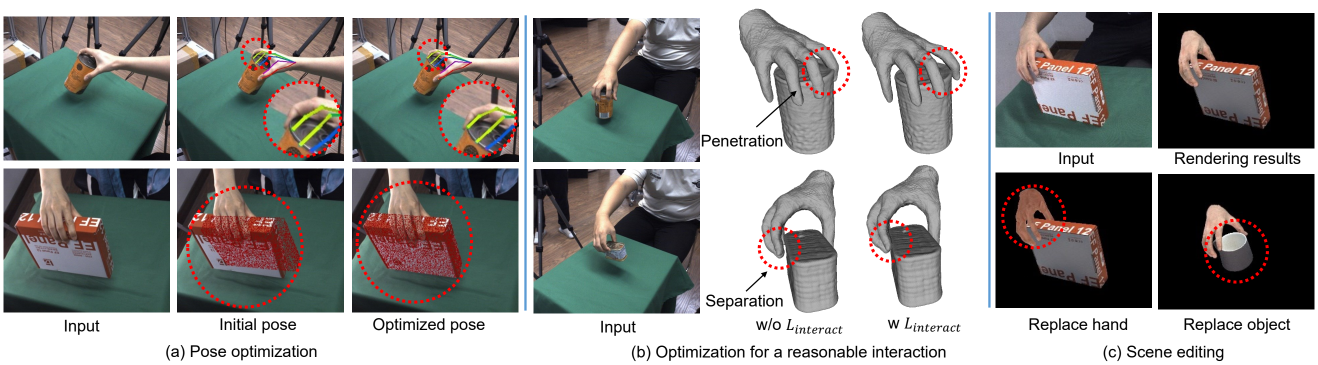


Figure 6: (a) Effect of pose optimization in online stage with joint model fitting. Optimization can improve the pose accuracy. (b) Effect of interaction loss. Interaction loss can facilitate to achieve reasonable hand-object interaction. (c) Editing of hand-object interaction scenes. We can replace the hand, object models and change the poses to get realistic rendering results.

Method	MPJPE ↓	AD ↓	ADD ↑	ADD-S ↑
CosyPose (CP) [27]	-	21.10	60.46	84.20
I2L [32]	18.39	-	-	-
I2L+CP+Mesh Fitting	20.60	21.10	60.54	84.25
GHPT [3]	13.28	-	-	-
GHPT+CP+Ours	10.80	**15.78**	**71.09**	**93.67**
LT [19]	9.66	-	-	-
LT+CP+A-NeRF	9.22	20.94	61.01	84.39
LT+CP+Ours	**9.09**	15.95	70.73	93.20

Table 1: Comparison on pose estimaion with SoTA methods. High-quality rendering results with our method are conducive to obtaining more accurate poses in the rendering-based optimization.

methods including LT [19] and GHPT [3], a single view pose estimation method I2L [32] and a fitting method based on A-NeRF [45]. In object 6D pose estimation, we compare with the SoTA multi-view method CosyPose [27]. During comparison, we optimize the pose initialized by LT, GHPT and CosyPose, and replace our hand model with A-NeRF based hand model for pose refinement (i.e. 'LT+CP+A-NeRF'). We also compare with the mesh-based fitting methods. I2L predicts the MANO parameters, and we use the untextured MANO hand mesh and the object mesh obtained in the offline stage to optimize the pose by fitting without color loss (i.e. 'I2L+CP+Mesh Fitting'). We show the results on HandObject under eight views in Table 1. We also test the accuracy of pose estimation under different number views on HandObject and Synthetic DexYCB. Table 3 shows the hand pose estimation results compared with LT, and the qualitative comparisons are demonstrated in the first row of Fig. 6(a). Table 2 shows object pose estimation results compared with CosyPose, and the qualitative comparisons are shown in the second row of Fig. 6(a). We observe

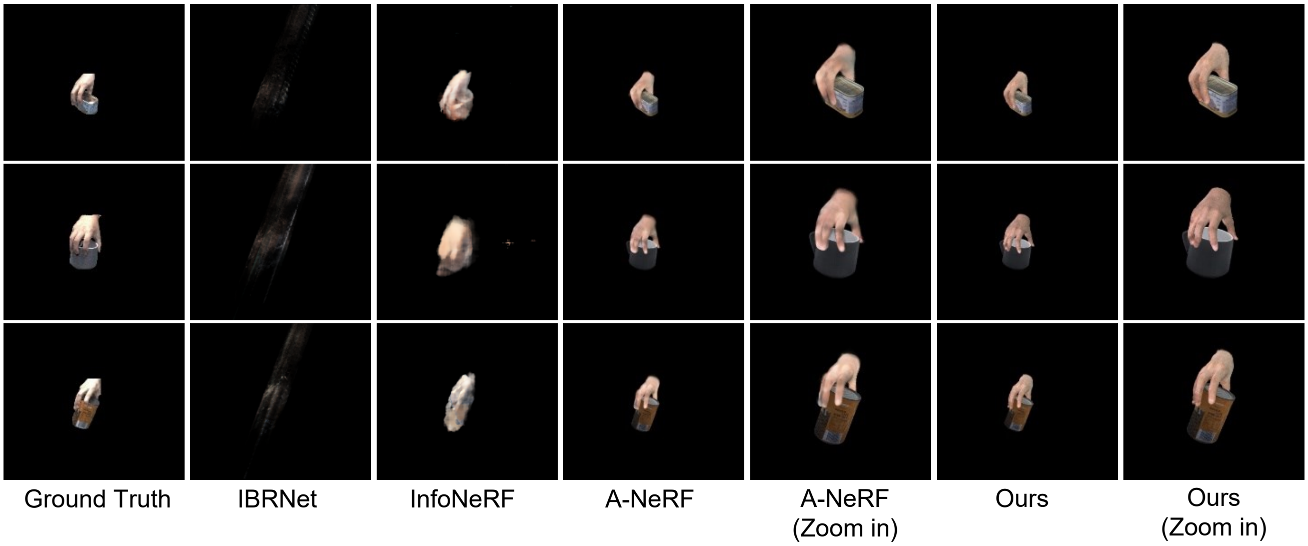


Figure 7: Rendering quality comparison on the HandObject dataset. We zoom in the rendering results for demonstration. Our method can achieve high-quality rendering results. Compared with A-NeRF [45] based hand model, our method preserves more realistic details.

	Object	8 views						6 views						3 views					
		CosyPose [27]			Ours			CosyPose [27]			Ours			CosyPose [27]			Ours		
		AD ↓	ADD ↑	ADD-S ↑	AD ↓	ADD ↑	ADD-S ↑	AD ↓	ADD ↑	ADD-S ↑	AD ↓	ADD ↑	ADD-S ↑	AD ↓	ADD ↑	ADD-S ↑	AD ↓	ADD ↑	ADD-S ↑
Synthetic DexYCB	002-master-chef-can	10.40	77.48	100.00	**5.56**	**90.73**	100.00	12.48	65.56	100.00	**8.23**	**80.79**	100.00	10.71	90.07	100.00	**7.08**	**95.36**	100.00
	003-cracker-box	15.15	70.95	89.86	**3.57**	**97.97**	**100.00**	16.98	60.81	91.89	**5.94**	**90.54**	**100.00**	15.98	72.97	89.86	**6.73**	**93.24**	**97.30**
	006-mustard-bottle	45.10	6.54	73.20	**31.93**	**53.59**	**97.39**	48.81	13.73	60.13	**35.63**	**43.14**	**87.58**	45.99	7.84	67.97	**34.37**	**38.56**	**91.50**
	010-potted-meat-can	27.58	23.02	90.65	**11.38**	**76.98**	**100.00**	37.47	8.63	84.89	**26.95**	**48.92**	**99.28**	28.27	28.06	78.42	**15.88**	**69.78**	**99.28**
	011-banana	29.29	36.55	60.00	**13.53**	**69.66**	**95.86**	33.91	26.90	51.72	**19.17**	**45.52**	**87.59**	46.85	22.07	42.76	**44.01**	**37.24**	**73.10**
Hand-Object	bean-can	19.43	62.65	90.62	**16.86**	**66.83**	**93.80**	23.84	49.92	87.10	**20.91**	**57.45**	**91.12**	21.46	58.63	87.94	**21.06**	**59.30**	**88.44**
	box	20.59	59.39	87.43	**14.95**	**76.55**	**95.57**	23.47	51.65	80.10	**18.35**	**66.16**	**91.54**	35.44	26.35	53.26	**32.75**	**33.28**	**59.31**
	cup	25.72	52.07	74.01	**17.79**	**61.76**	**90.13**	18.10	60.53	92.25	**14.18**	**76.04**	**96.74**	27.19	33.92	73.39	**24.76**	**39.03**	**79.82**
	meat-can	15.35	75.64	90.13	**13.74**	**79.14**	**93.47**	15.21	71.97	92.52	**13.71**	**77.23**	**94.11**	21.03	47.29	86.46	**19.56**	**54.14**	**89.49**

Table 2: Comparison of object pose estimation under different camera views on HandObject and Synthetic DexYCB.

that our method outperforms the SoTA methods. Benefiting from the pre-built models, our method can exploit more dense supervision on image pixels than sparse keypoint supervision in LT, I2L, GHPT and CosyPose. Compared with A-NeRF hand model based fitting, our high-quality rendering models are conducive to obtaining more accurate poses. Compared with untextured mesh based fitting, we find that it is inferior to the color loss to provide sufficient constraints to achieve accurate pose.

	Object	8 views		6 views		3 views	
		LT [19]	Ours	LT [19]	Ours	LT [19]	Ours
Synthetic DexYCB	002-master-chef-can	9.61	**7.84**	10.08	**8.21**	15.57	**13.65**
	003-cracker-box	10.72	**9.87**	12.09	**11.25**	12.06	**11.03**
	006-mustard-bottle	9.15	**7.73**	10.22	**8.58**	10.74	**9.31**
	010-potted-meat-can	8.27	**7.08**	8.68	**7.62**	12.06	**10.62**
	011-banana	11.21	**10.53**	11.67	**10.98**	13.50	**13.47**
Hand-Object	bean-can	8.85	**8.07**	9.31	**8.17**	14.05	**11.98**
	box	9.63	**9.29**	10.25	**9.89**	17.81	**16.73**
	cup	10.51	**9.89**	11.12	**10.45**	16.19	**15.29**
	meat-can	8.97	**8.20**	9.46	**8.50**	15.33	**13.50**

Table 3: MPJPE (mm) of hand pose estimation under different view numbers on HandObject and Synthetic DexYCB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
IBRNet [49]	17.02	73.75	0.291
InfoNeRF [25]	18.70	87.92	0.161
A-NeRF [45]	21.99	93.40	0.066
Ours	**22.20**	**93.71**	**0.059**

Table 4: Quantitative comparison of rendering quality.

Rendering Quality. We compare the rendering quality in hand-object interaction scenes with A-NeRF [45], IBR-Net [49] and InfoNeRF [25] on HandObject dataset under five test views. Table 4 shows quantitative comparison, and Fig. 7 shows qualitative comparison. We replace our hand model with A-NeRF based hand model and use the same object model for fitting and rendering (i.e. 'A-NeRF'). The rendering results with our method perform better than others, because our offline models provide strong shape and appearance priors which are more suitable for few-shot neural rendering. Compared to the density representation in A-

	Object	$\mathcal{L}_{render}+\mathcal{L}_{pose}$				$\mathcal{L}_{render}+\mathcal{L}_{pose}+\mathcal{L}_{interact}$			
		MPJPE ↓	AD ↓	Int-Vol ↓	Pen-Dep ↓	MPJPE↓	AD ↓	Int-Vol ↓	Pen-Dep ↓
Synthetic DexYCB	002-master-chef-can	8.60	6.06	8.54	9.65	**7.84**	**5.56**	**4.87**	**6.09**
	003-cracker-box	9.97	4.43	5.84	13.09	**9.87**	**3.57**	**3.60**	**6.36**
	006-mustard-bottle	8.16	32.71	6.68	8.73	**7.73**	**31.93**	**2.07**	**3.08**
	010-potted-meat-can	7.19	12.09	4.39	9.22	**7.08**	**11.38**	**1.58**	**1.88**
	011-banana	10.71	14.19	2.17	5.34	**10.53**	**13.53**	**0.93**	**1.54**
Hand-Object	bean-can	**7.72**	16.92	4.95	6.28	8.07	**16.86**	**3.17**	**3.52**
	box	**9.23**	15.18	6.91	11.12	9.29	**14.95**	**5.79**	**8.68**
	cup	**9.77**	17.89	5.00	9.14	9.89	**17.79**	**4.26**	**6.82**
	meat-can	**7.90**	13.81	4.26	6.50	8.20	**13.74**	**3.23**	**3.85**

Table 5: Effect of interaction loss on Interpenetration level.

Datasets	$\mathcal{L}_{fit}$			$\mathcal{L}_{fit}+\mathcal{L}_{smo}$			$\mathcal{L}_{fit}+\mathcal{L}_{smo}+\mathcal{L}_{stable}$		
	Acc-J↓	Acc-V↓	PCI↑	Acc-J↓	Acc-V↓	PCI↑	Acc-J↓	Acc-V↓	PCI↑
Synthetic DexYCB	8.04	26.76	10.98	**6.78**	19.51	17.38	7.51	**19.18**	**35.02**
HandObject	6.28	10.07	29.68	**6.20**	6.35	36.37	6.25	**6.10**	**51.18**

Table 6: Effect of smooth loss and stable contact loss. Smooth loss will reduce pose jitters with lower acceleration error, and stable contact loss will make the contact area more stable with higher PCI.

NeRF, the SDF representation in our model makes the shape sharper.

4.4. Ablation Study

Effect of Interaction Loss. We compare the interpenetration level as shown in Table 5 and Fig. 6(b). We observe that our model with interaction loss can achieve excellent performance on Penetration depth (Pen-Dep) and Intersection volume (Int-Vol), i.e., while only sacrificing negligible performance drop in hand pose. We also compare the rendering quality on the HandObject dataset as shown in Table 7. After fitting (i.e. '$\mathcal{L}_{render}+\mathcal{L}_{pose}$', '$\mathcal{L}_{render}+\mathcal{L}_{pose}+\mathcal{L}_{interact}$'), the rendering quality becomes better. The interaction loss $\mathcal{L}_{interact}$ can further improve the rendering quality, because the incorrect color caused by unreasonable interactions such as penetration can be reduced by the loss.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
w/o $\mathcal{L}_{fit}$	22.07	93.32	0.0622
$\mathcal{L}_{render}+\mathcal{L}_{pose}$	22.17	93.54	0.0607
$\mathcal{L}_{render}+\mathcal{L}_{pose}+\mathcal{L}_{interact}$	**22.25**	**93.57**	**0.0605**

Table 7: Effect of interaction loss on rendering quality.

Effect of Smooth Loss and Stable Contact Loss. Table 6 shows the results on acceleration error and PCI. Compared with applying $\mathcal{L}_{fit}$ only, we observe that the smooth loss can reduce pose jitters and lead to smoother pose change with lower acceleration error. After adding stable contact loss, the PCI values increase, indicating the contact regions tend to be stable, and acceleration error on object pose is also reduced, indicating a further reduction in object jitters.

4.5. Hand-Object Interaction Editing

Our model can be driven by controllable variables such as hand pose $\mathbf{J}$, object pose $\mathbf{R}_o$ and $\mathbf{T}_o$. We can edit hand-object interaction scenes, including replacing hand or object models, and poses as shown in Fig. 6(c).

5. Conclusion

We propose a novel neural rendering and pose estimation system for hand-object interaction from sparse view images. We design a two-stage approach (i.e. offline model building and online model fitting) to achieve accurate hand-object pose estimation and photo-realistic novel view synthesis. We utilize effective geometric constraints to conduct rendering-based online model fitting. Various experiments demonstrate that our method outperforms the SoTA pose estimation and few-shot neural rendering methods. In the future work, we will take into account lighting conditions to reduce unrealistic results caused by shadows from different illuminations and improve the efficiency of our method with Instant-NGP [33].

Acknowledgments. This work was supported in part by National Key R&D Program of China (2022ZD0117900).

References

[1] Sridhar Pandian Arunachalam, Sneha Silwal, Ben Evans, and Lerrel Pinto. Dexterous imitation made easy: A learning-based framework for efficient dexterous manipulation. *arXiv preprint arXiv:2203.13251*, 2022.

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.

[3] Kristijan Bartol, David Bojanić, Tomislav Petković, and Tomislav Pribanić. Generalizable human pose triangulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11028–11037, 2022.

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

[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.

[6] Xu Chen, Yufeng Zheng, Michael J Black, Otmar Hilliges, and Andreas Geiger. Snarf: Differentiable forward skinning for animating non-rigid neural implicit shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11594–11604, 2021.

[7] Zerui Chen, Yana Hasson, Cordelia Schmid, and Ivan Laptev. Alignsdf: Pose-aligned signed distance fields for hand-object reconstruction. In *European Conference on Computer Vision*, pages 231–248. Springer, 2022.

[8] Enric Corona, Tomas Hodan, Minh Vo, Francesc Moreno-Noguer, Chris Sweeney, Richard Newcombe, and Lingni Ma. Lisa: Learning implicit shape and appearance of hands. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20533–20543, 2022.

[9] Boyang Deng, John P Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Nasa neural articulated shape approximation. In *Proceedings of the European Conference on Computer Vision*, pages 612–628, 2020.

[10] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[11] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *International Conference on Machine Learning*, pages 3789–3799, 2020.

[12] Michelle Guo, Alireza Fathi, Jiajun Wu, and Thomas Funkhouser. Object-centric neural scene rendering. *arXiv preprint arXiv:2012.08503*, 2020.

[13] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3196–3206, 2020.

[14] Shreyas Hampali, Sayan Deb Sarkar, Mahdi Rad, and Vincent Lepetit. Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11090–11100, 2022.

[15] Shangchen Han, Po-chen Wu, Yubo Zhang, Beibei Liu, Linguang Zhang, Zheng Wang, Weiguang Si, Peizhao Zhang, Yujun Cai, Tomas Hodan, et al. Umetrack: Unified multi-view end-to-end hand tracking for vr. In *SIGGRAPH Asia*, pages 1–9, 2022.

[16] 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.

[17] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11807–11816, 2019.

[18] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Asian Conference on Computer Vision*, pages 548–562, 2012.

[19] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7718–7727, 2019.

[20] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5885–5894, 2021.

[21] Hanwen Jiang, Shaowei Liu, Jiashun Wang, and Xiaolong Wang. Hand-object contact consistency reasoning for human grasps generation. *arXiv preprint arXiv:2104.03304*, 2021.

[22] Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5614–5623, 2019.

[23] Korrawe Karunratanakul, Adrian Spurr, Zicong Fan, Otmar Hilliges, and Siyu Tang. A skeleton-driven neural occupancy representation for articulated hands. In *International Conference on 3D Vision (3DV)*, pages 11–21. IEEE, 2021.

[24] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J Black, Krikamol Muandet, and Siyu Tang. Grasping field: Learning implicit representations for human grasps. In *International Conference on 3D Vision (3DV)*, pages 333–344. IEEE, 2020.

[25] Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12912–12921, 2022.

[26] Taein Kwon, Bugra Tekin, Jan Stuhmer, Federica Bogo, and Marc Pollefeys. H2o: Two hands manipulating objects for first person interaction recognition. *arXiv preprint arXiv:2104.11181*, 2021.

[27] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *Proceedings of the European Conference on Computer Vision*, pages 574–591, 2020.

[28] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*, 2019.

[29] Marko Mihajlovic, Aayush Bansal, Michael Zollhoefer, Siyu Tang, and Shunsuke Saito. Keypointnerf: Generalizing image-based volumetric avatars using relative spatial encoding of keypoints. In *European Conference on Computer Vision (ECCV 2022)*, 2022.

[30] Marko Mihajlovic, Yan Zhang, Michael J Black, and Siyu Tang. Leap: Learning articulated occupancy of people. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10461–10471, 2021.

[31] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision*, pages 405–421, 2020.

[32] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 752–768. Springer, 2020.

[33] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)*, 41(4):1–15, 2022.

[34] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021.

[35] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5762–5772, 2021.

[36] Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L Davidson, Sameh Khamis, Mingsong Dou, et al. Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the Annual Symposium on User Interface Software and Technology*, pages 741–754, 2016.

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

[38] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14314–14323, 2021.

[39] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9054–9063, 2021.

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

[41] Neng Qian, Jiayi Wang, Franziska Mueller, Florian Bernard, Vladislav Golyanik, and Christian Theobalt. Html: a parametric hand texture model for 3d hand reconstruction and personalization. In *Proceedings of the European Conference on Computer Vision*, pages 54–71, 2020.

[42] Yuzhe Qin, Hao Su, and Xiaolong Wang. From one hand to multiple hands: Imitation learning for dexterous manipulation from single-camera teleoperation. *IEEE Robotics and Automation Letters*, 7(4):10873–10881, 2022.

[43] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020.

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

[45] Shih-Yang Su, Frank Yu, Michael Zollhöfer, and Helge Rhodin. A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose. *Advances in Neural Information Processing Systems*, 34, 2021.

[46] Tze Ho Elden Tse, Kwang In Kim, Ales Leonardis, and Hyung Jin Chang. Collaborative learning for hand and object reconstruction with attention-guided graph convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1664–1674, 2022.

[47] Tze Ho Elden Tse, Zhongqun Zhang, Kwang In Kim, Ales Leonardis, Feng Zheng, and Hyung Jin Chang. S 2 contact: Graph-based network for 3d hand-object contact estimation with semi-supervised learning. In *European Conference on Computer Vision*, pages 568–584. Springer, 2022.

[48] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.

[49] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Pro-*

ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4690–4699, 2021.

[50] Shaofei Wang, Katja Schwarz, Andreas Geiger, and Siyu Tang. Arah: Animatable volume rendering of articulated human sdfs. *arXiv preprint arXiv:2210.10036*, 2022.

[51] Chung-Yi Weng, Brian Curless, Pratul P Srinivasan, Jonathan T Barron, and Ira Kemelmacher-Shlizerman. Humannerf: Free-viewpoint rendering of moving people from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16210–16220, 2022.

[52] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[53] Bangbang Yang, Yinda Zhang, Yijin Li, Zhaopeng Cui, Sean Fanello, Hujun Bao, and Guofeng Zhang. Neural rendering in a room: amodal 3d understanding and free-viewpoint rendering for the closed scene composed of pre-captured objects. *ACM Transactions on Graphics (TOG)*, 41(4):1–10, 2022.

[54] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13779–13788, 2021.

[55] 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.

[56] Guangming Yao, Hongzhi Wu, Yi Yuan, and Kun Zhou. Learning implicit body representations from double diffusion based neural radiance fields. *arXiv preprint arXiv:2112.12390*, 2021.

[57] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34, 2021.

[58] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33:2492–2502, 2020.

[59] Yufei Ye, Abhinav Gupta, and Shubham Tulsiani. What's in your hands? 3d reconstruction of generic objects in hands. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3895–3905, 2022.

[60] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021.

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

[62] Hao Zhang, Yuxiao Zhou, Yifei Tian, Jun-Hai Yong, and Feng Xu. Single depth view based real-time reconstruction of hand-object interactions. *ACM Transactions on Graphics (TOG)*, 40(3):1–12, 2021.

[63] Siwei Zhang, Yan Zhang, Qianli Ma, Michael J Black, and Siyu Tang. Place: Proximity learning of articulation and contact in 3d environments. In *International Conference on 3D Vision (3DV)*, pages 642–651, 2020.

[64] Zerong Zheng, Han Huang, Tao Yu, Hongwen Zhang, Yandong Guo, and Yebin Liu. Structured local radiance fields for human avatar modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15893–15903, 2022.

[65] 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.